Module 2

Initial Value Problems

2.4 Implicit method

2.5 Stiffness

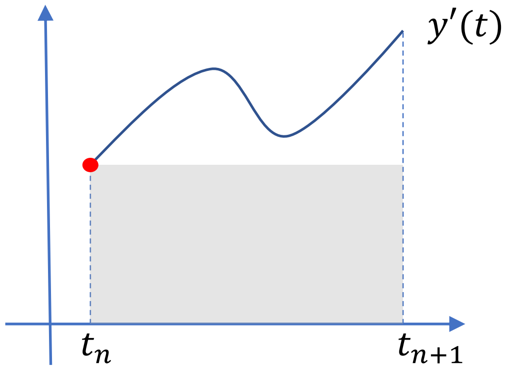
2.6 Linear Multistep Methods

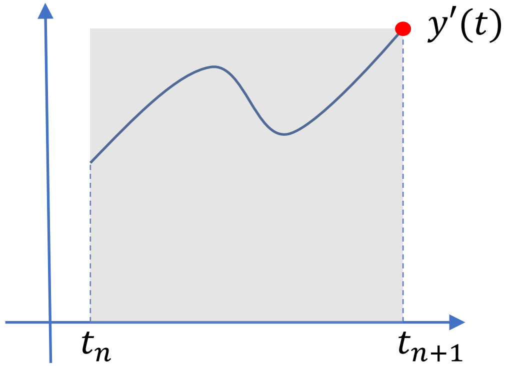
Akash Anand
MATH, IIT KANPUR

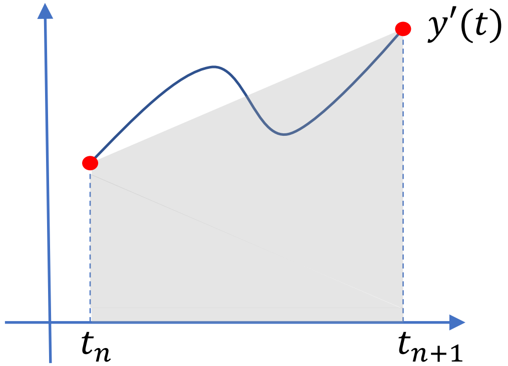
Can we make the method higher order?

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?





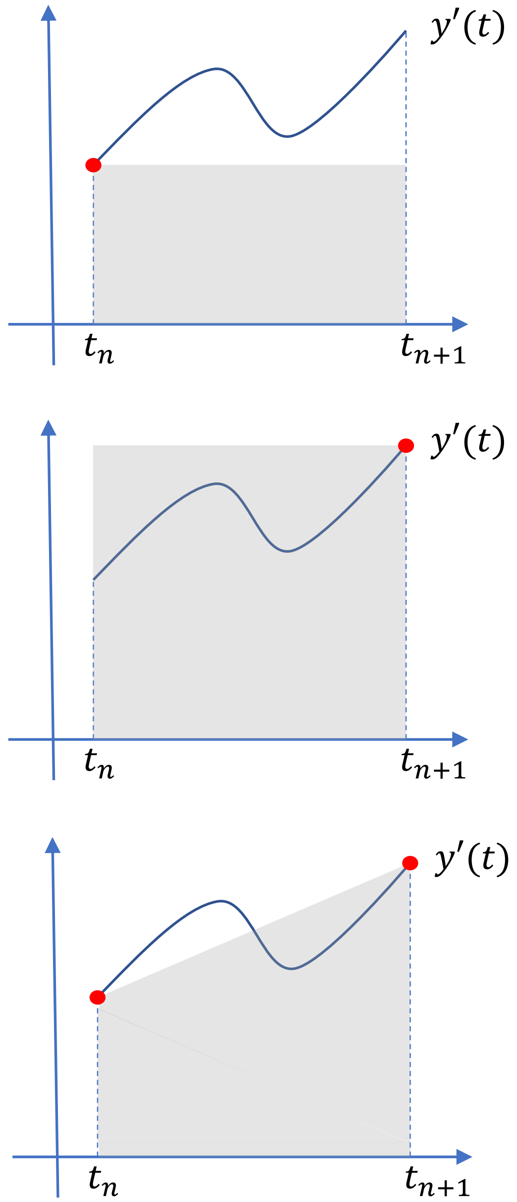


Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

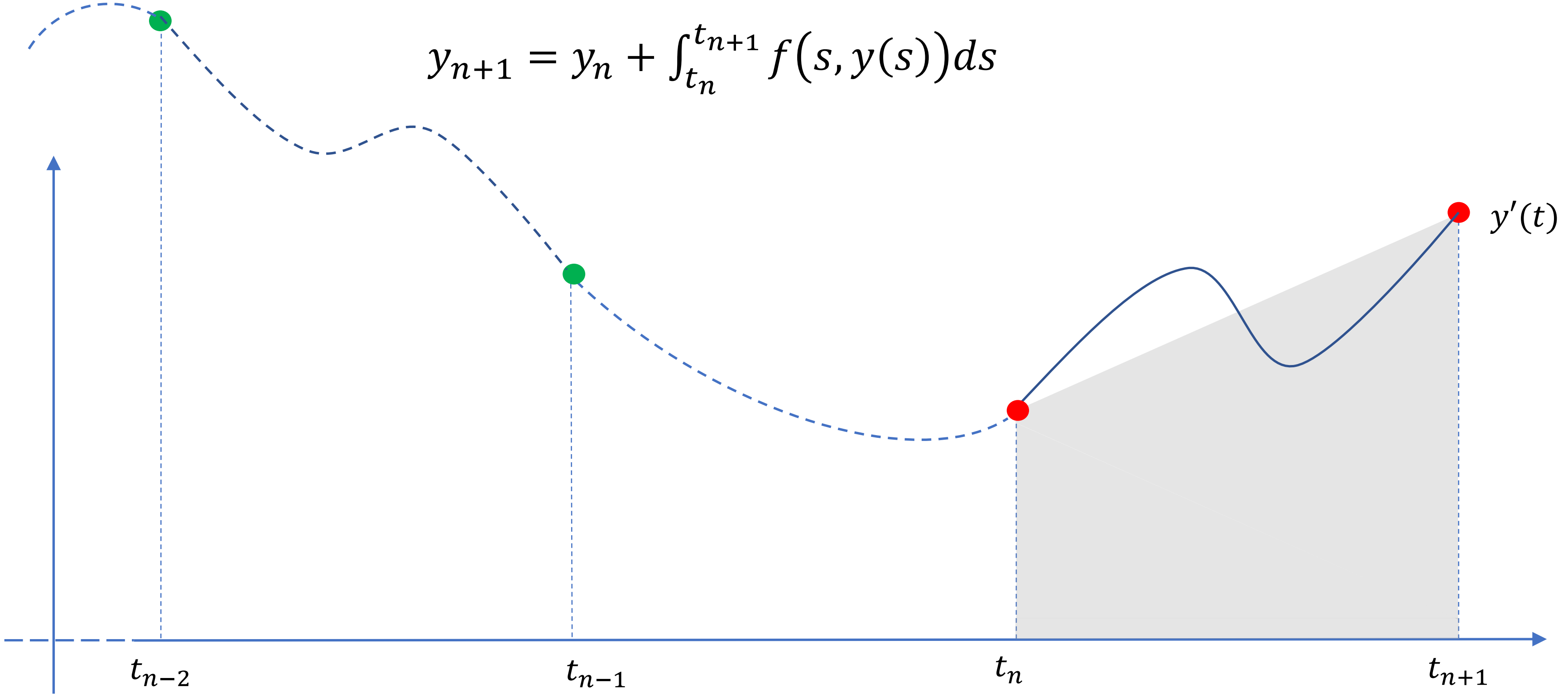
Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.



Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds$$



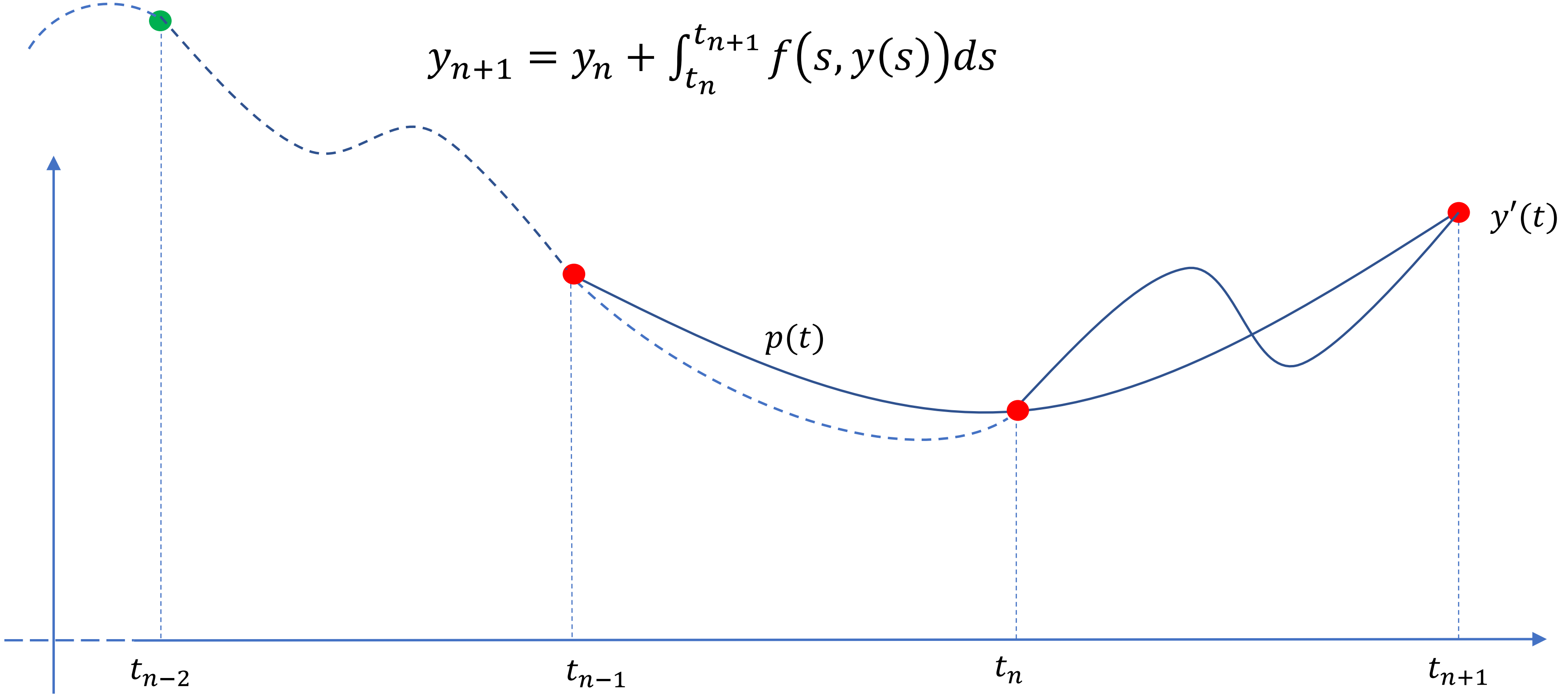
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds$$



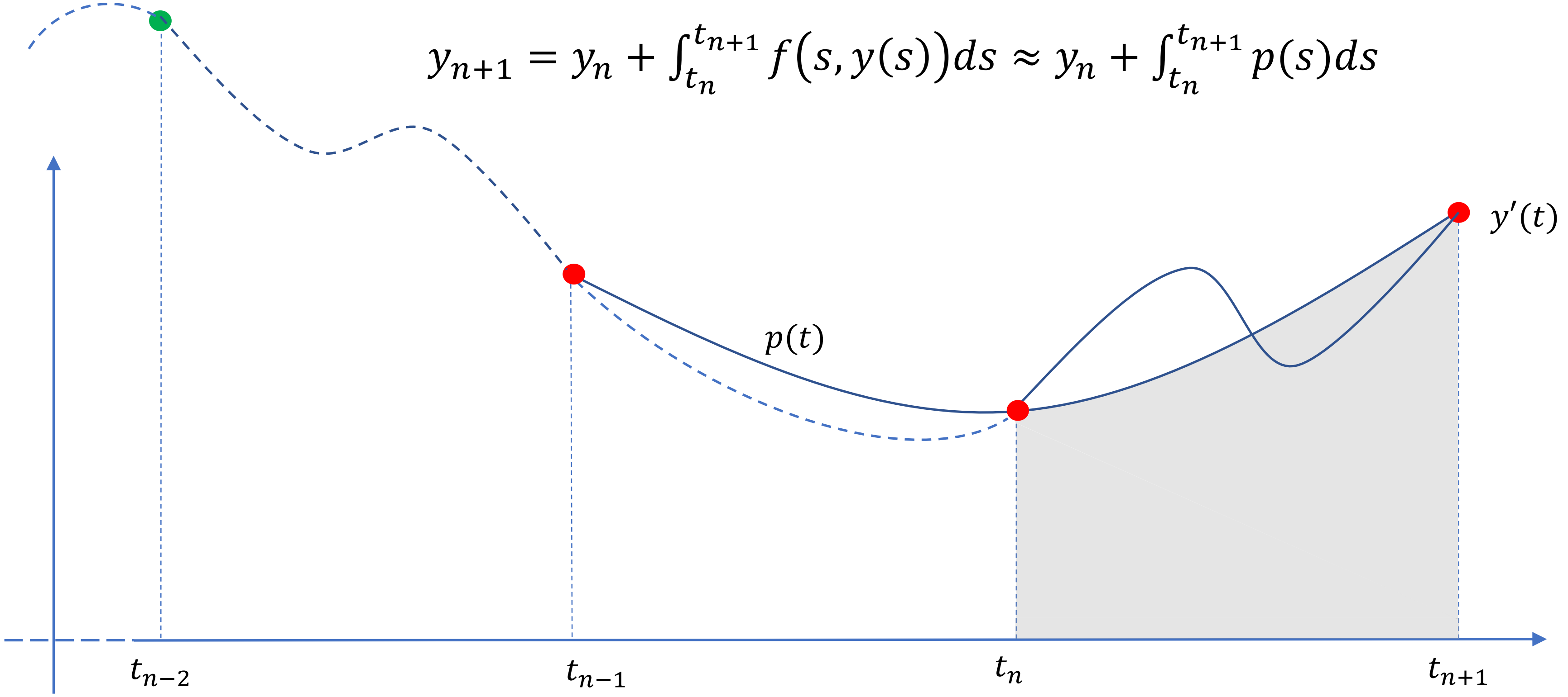
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds$$



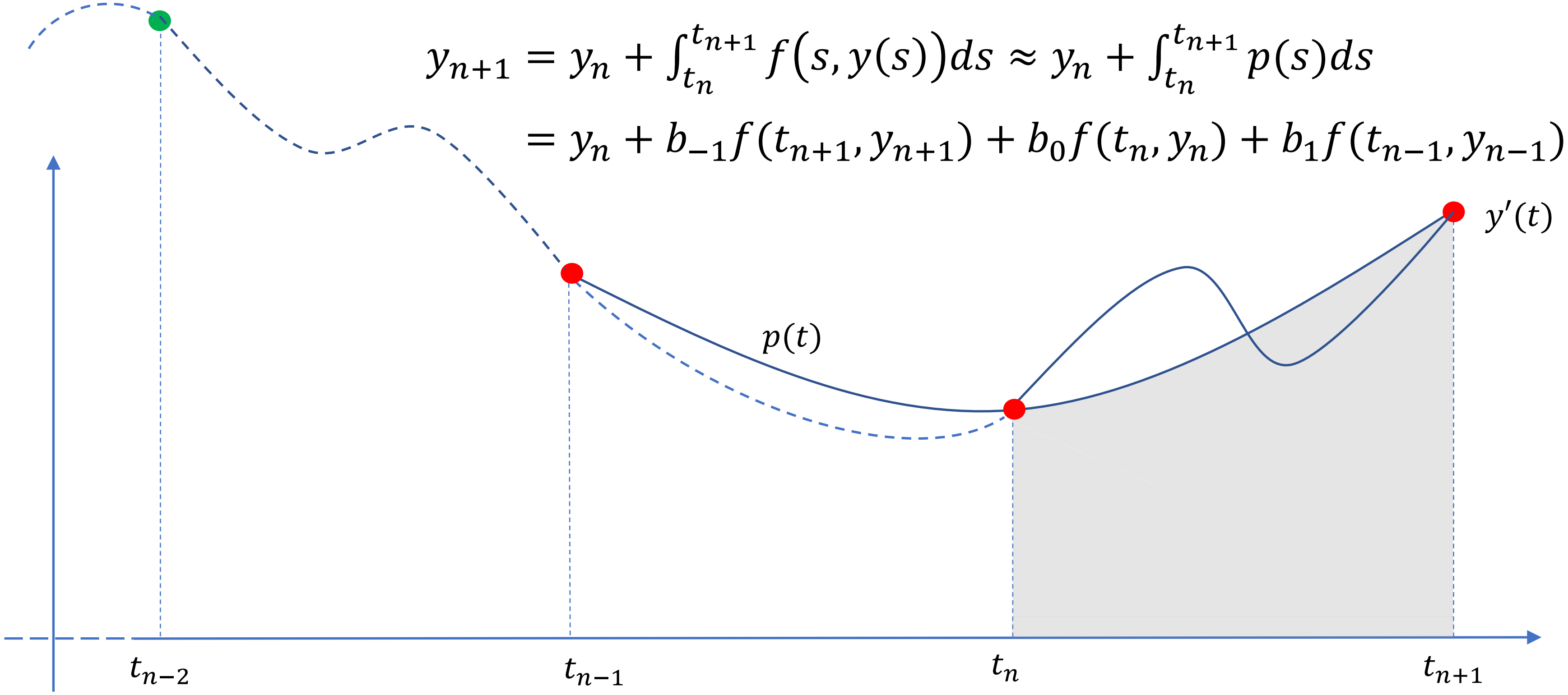
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$
\begin{aligned}
y_{n+1} &= y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds \\
&= y_n + b_{-1} f(t_{n+1}, y_{n+1}) + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1})
\end{aligned}
$$



Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

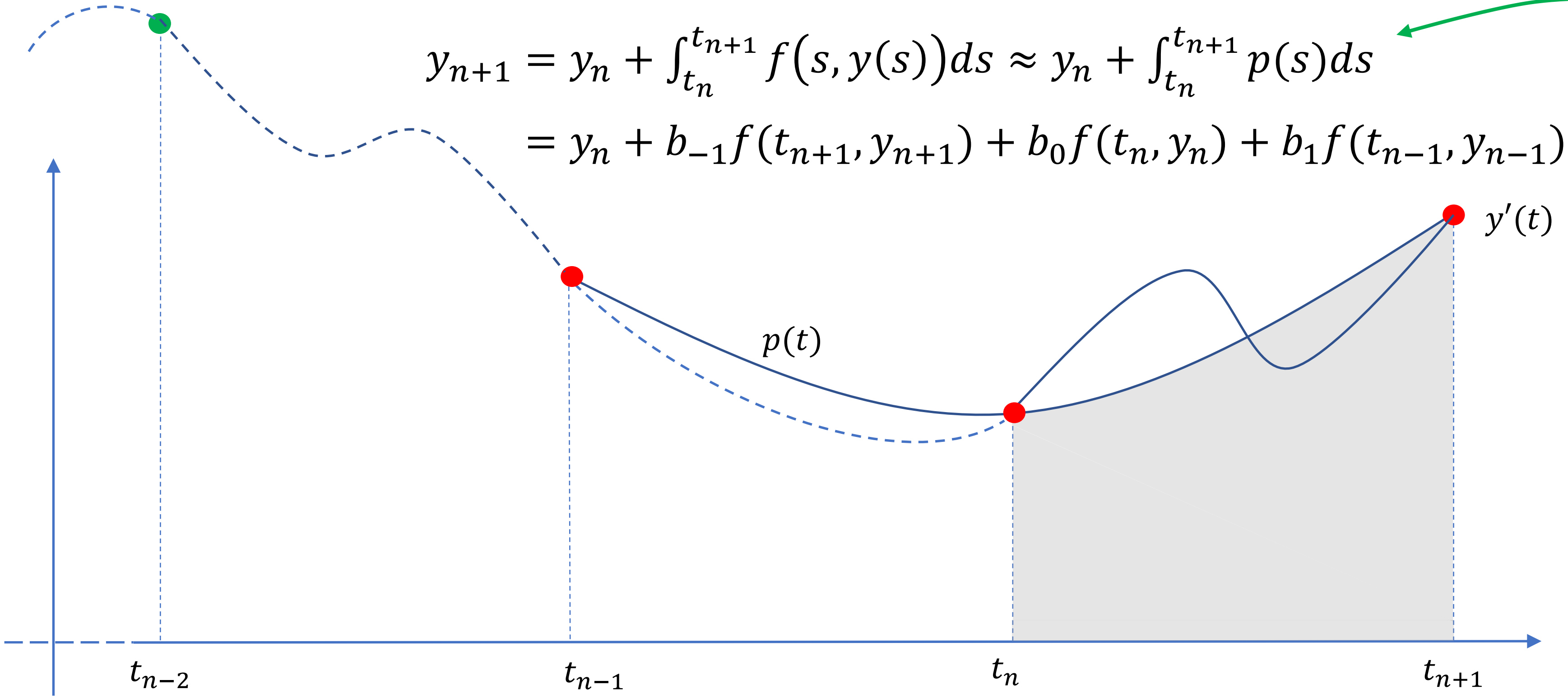
Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds$$

$$= y_n + b_{-1} f(t_{n+1}, y_{n+1}) + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1})$$

Adams—Moulton method



Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

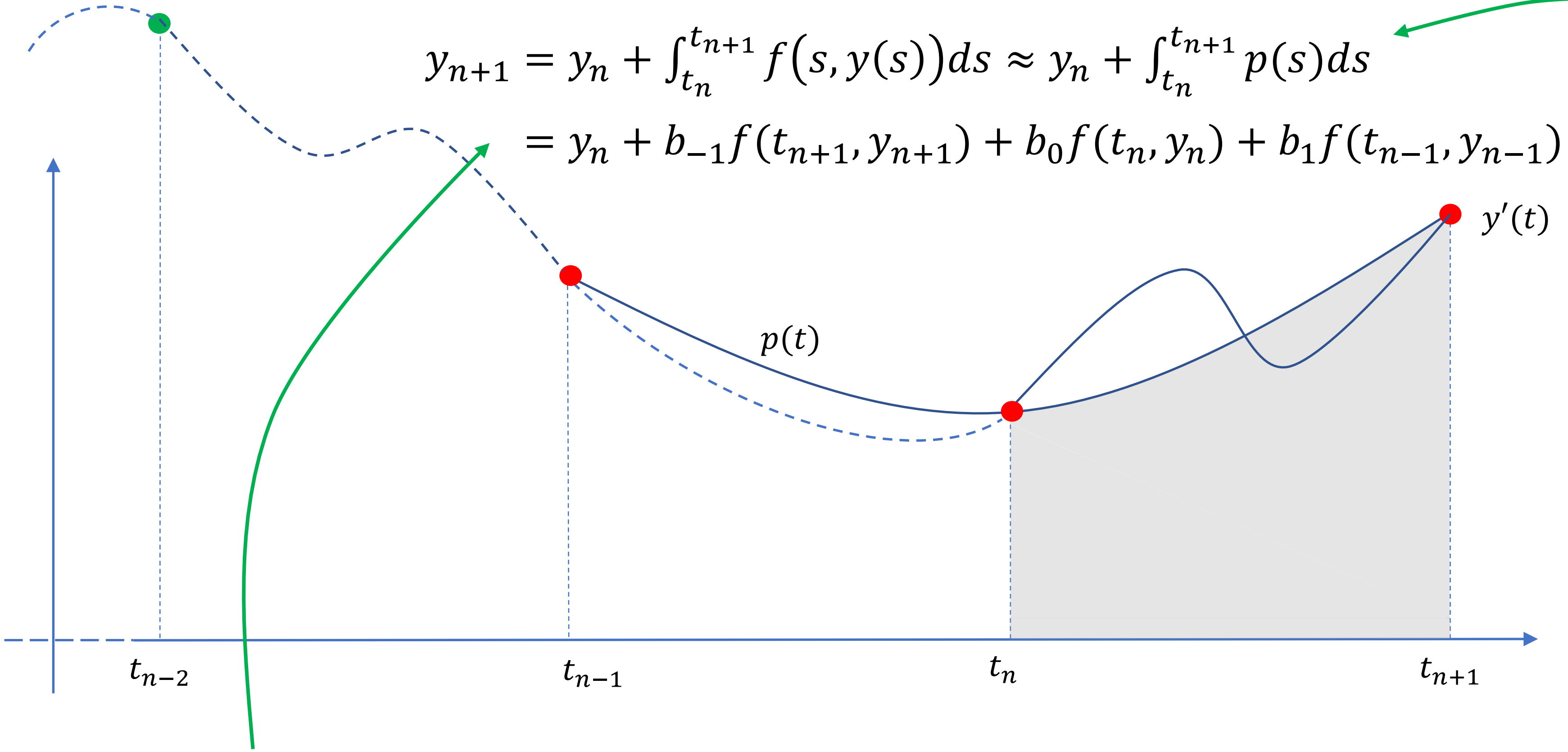
Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds$$

$$= y_n + b_{-1} f(t_{n+1}, y_{n+1}) + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1})$$

Adams—Moulton method



An implicit method

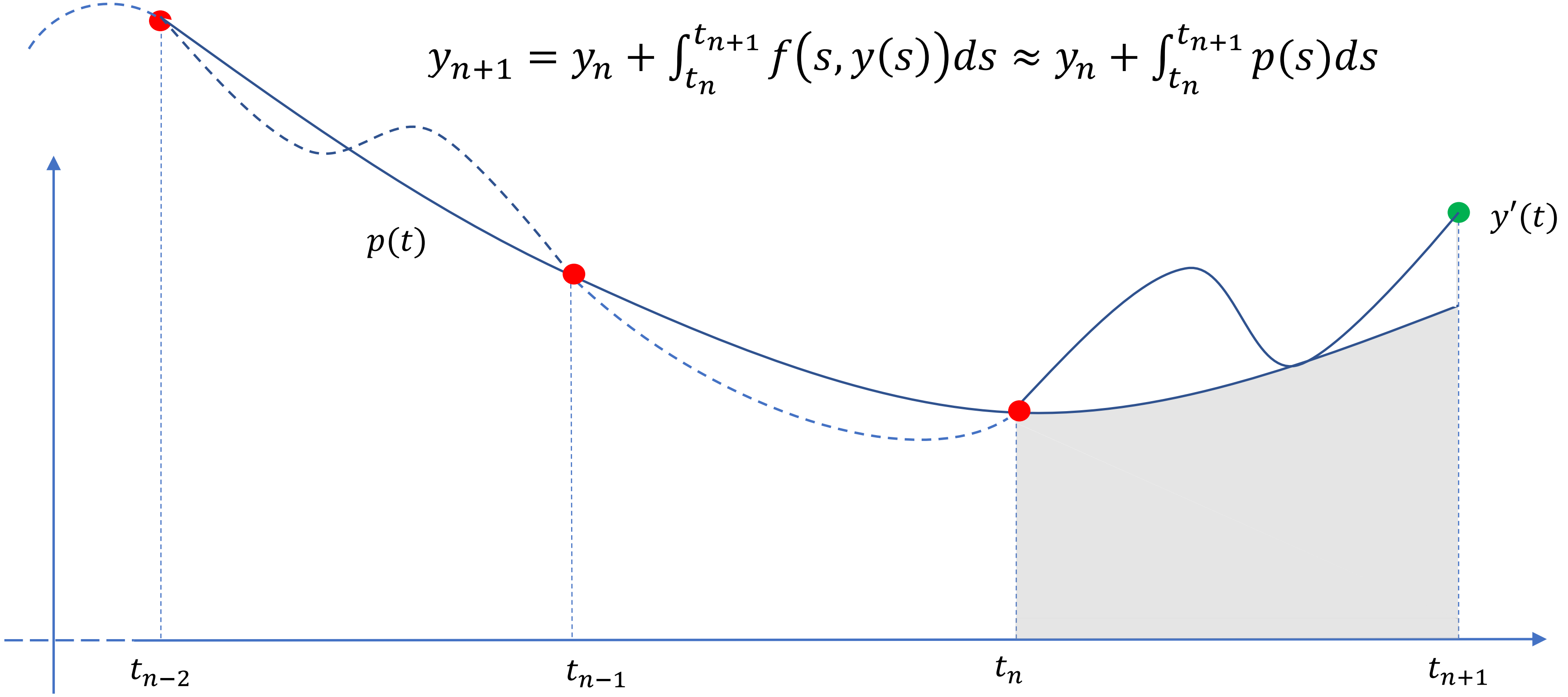
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds$$



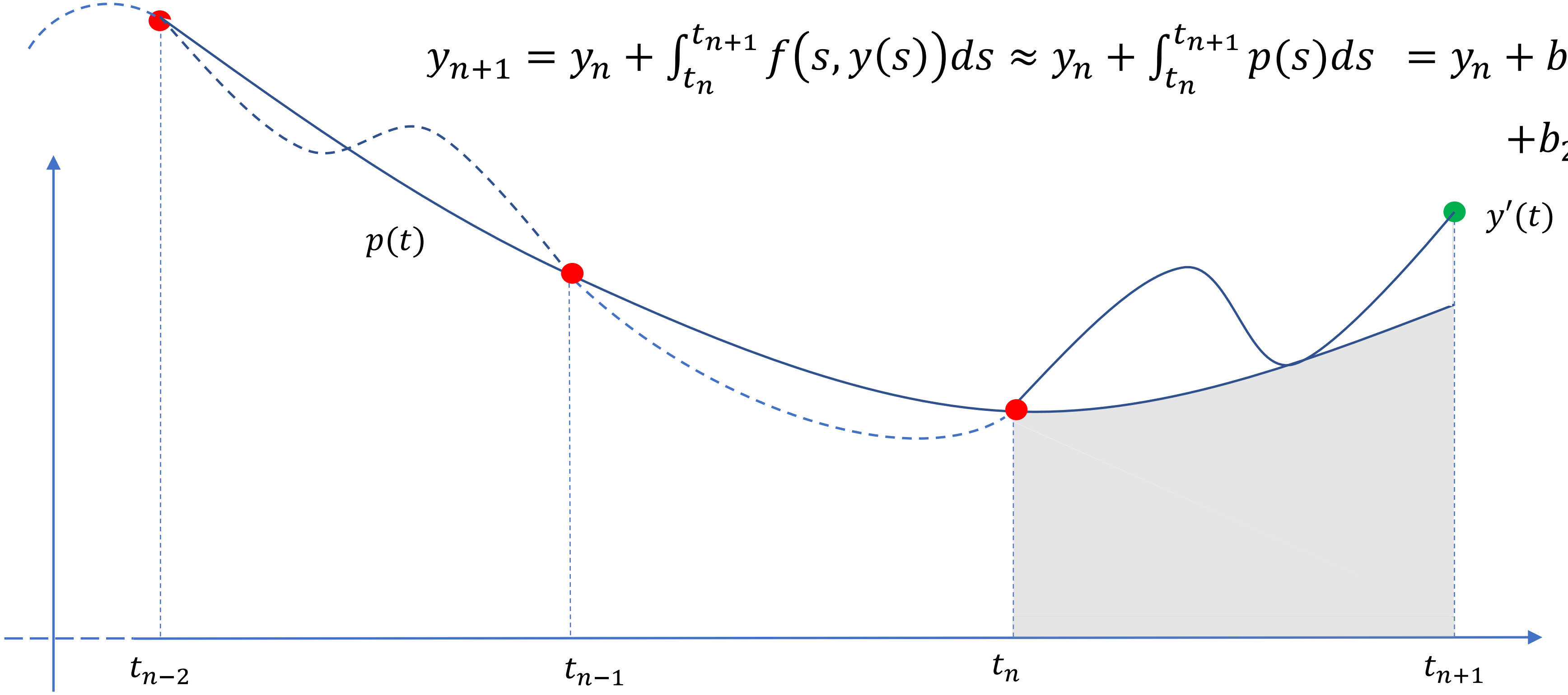
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds \ = y_n + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1}) + b_2 f(t_{n-2}, y_{n-2})$$



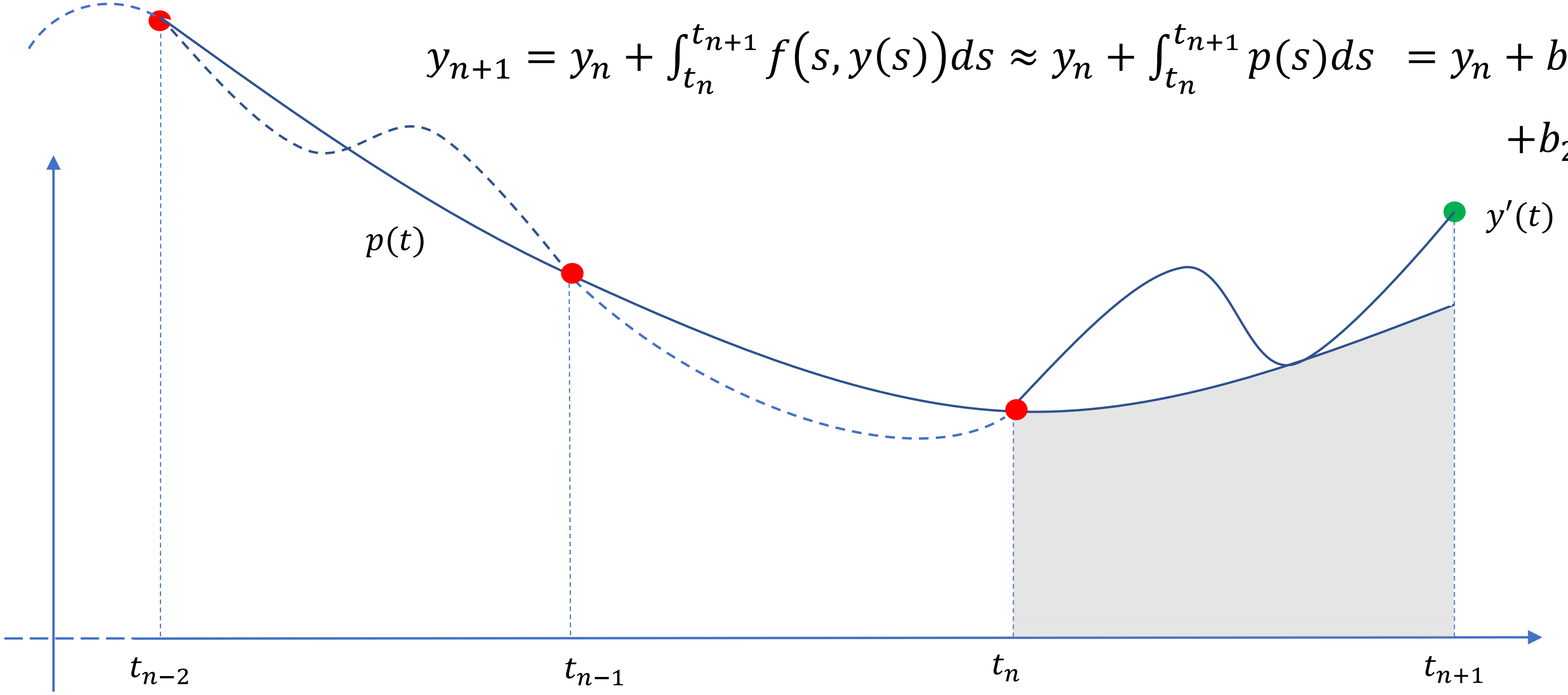
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Can we make the method higher order?

Following the idea that we used to derive trapezoidal method, we could use higher order polynomial approximations.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\big(s, y(s)\big)ds \approx y_n + \int_{t_n}^{t_{n+1}} p(s)ds \;= y_n + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1}) + b_2 f(t_{n-2}, y_{n-2})$$



Adams-Bashford method

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

… we use some other method such as a single step method.

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

… we use some other method such as a single step method.

Examples:

1. $y_{n+1} = y_n + hf(t_n, y_n)$ *… an explicit one step method*
2. $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ *… an implicit one step method*
3. $y_{n+1} = y_n + h\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)/2$ *… an implicit one step method*

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

… we use some other method such as a single step method.

Examples:

1. $y_{n+1} = y_n + hf(t_n, y_n)$ … *an explicit one step method*
2. $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ … *an implicit one step method*
3. $y_{n+1} = y_n + h\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)/2$ … *an implicit one step method*
4. $y_{n+1} = y_n + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1}) + b_2 f(t_{n-2}, y_{n-2})$ … *an explicit three step method*

Initial Value Problems: Linear Multistep Methods

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

… we use some other method such as a single step method.

Examples:

1. $y_{n+1} = y_n + hf(t_n, y_n)$ … an explicit one step method
2. $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ … an implicit one step method
3. $y_{n+1} = y_n + h\,(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))/2$ … an implicit one step method
4. $y_{n+1} = y_n + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1}) + b_2 f(t_{n-2}, y_{n-2})$ … an explicit three step method
5. $y_{n+1} = y_n + h\,(f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n)))/2$ … an explicit one step method

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

… we use some other method such as a single step method.

Improved Euler Method

Examples:

1. $y_{n+1} = y_n + hf(t_n, y_n)$ *… an explicit one step method*
2. $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ *… an implicit one step method*
3. $y_{n+1} = y_n + h\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$ *… an implicit one step method*
4. $y_{n+1} = y_n + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1}) + b_2 f(t_{n-2}, y_{n-2})$ *… an explicit three step method*
5. $y_{n+1} = y_n + h\left(f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n))\right)/2$ *… an explicit one step method*

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider methods that take constant step size h and determine y_{n+1} using the values from several preceding steps:

$$y_{n+1} = \Phi(f, t_n, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k}, h).$$

Here y_{n+1} depends on $k+1$ previous values, so this is called a $(k+1)$-step method.

For $k \geq 1$, that is, for a 2 or more step method, how do we start the time marching? Note that we need to know $y_0, \dots, y_k$, to compute y_{k+1} and we typically only know y_0!

… we use some other method such as a single step method.

Improved Euler Method

Examples:

1. $y_{n+1} = y_n + hf(t_n, y_n)$ … an explicit one step method
2. $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ … an implicit one step method
3. $y_{n+1} = y_n + h\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$ … an implicit one step method
4. $y_{n+1} = y_n + b_0 f(t_n, y_n) + b_1 f(t_{n-1}, y_{n-1}) + b_2 f(t_{n-2}, y_{n-2})$ … an explicit three step method
5. $y_{n+1} = y_n + h\left(f(t_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n))\right)/2$ … an explicit one step method

Φ Is linear in y_n, $f(t_n, y_n)$, $f(t_{n+1}, y_{n+1})$, etc.

non-linear Φ

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider linear multistep methods with constant step size, which by definition, are methods of the form

$$y_{n+1} = -\mathrm{a}_0\mathrm{y}_\mathrm{n} - \mathrm{a}_1\mathrm{y}_{\mathrm{n}-1} - \cdots - a_k y_{n-k} + h[b_{-1}f_{n+1} + b_0 f_n + \cdots + b_k f_{n-k}]$$

where f_n *denotes* $f(t_n, y_n)$ *(for brevity) and* a_j*,* b_j *are constants which must be given and determine the specific method.*

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

We consider linear multistep methods with constant step size, which by definition, are methods of the form

$$y_{n+1} = -\mathrm{a}_0\mathrm{y}_\mathrm{n} - \mathrm{a}_1\mathrm{y}_{\mathrm{n}-1} - \cdots - a_k y_{n-k} + h[b_{-1}f_{n+1} + b_0 f_n + \cdots + b_k f_{n-k}]$$

where f_n *denotes* $f(t_n, y_n)$ *(for brevity) and* a_j, b_j *are constants which must be given and determine the specific method.*

For explicit linear multistep method, $b_{-1} = 0$.

Initial Value Problems: Linear Multistep Methods

We consider linear multistep methods with constant step size, which by definition, are methods of the form

$$y_{n+1} = -\mathrm{a}_0 \mathrm{y}_\mathrm{n} - \mathrm{a}_1 \mathrm{y}_{\mathrm{n}-1} - \cdots - a_k y_{n-k} + h[b_{-1} f_{n+1} + b_0 f_n + \cdots + b_k f_{n-k}]$$

where f_n *denotes* $f(t_n, y_n)$ *(for brevity) and* a_j, b_j *are constants which must be given and determine the specific method.*

For explicit linear multistep method, $b_{-1} = 0$.

It is also convenient to define $a_{-1} = 1$*, so that the method can we written more concisely as*

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f_{n-j} \, .$$

We consider linear multistep methods with constant step size, which by definition, are methods of the form

$$y_{n+1} = -\mathrm{a}_0 \mathrm{y}_\mathrm{n} - \mathrm{a}_1 \mathrm{y}_{\mathrm{n}-1} - \cdots - a_k y_{n-k} + h[b_{-1} f_{n+1} + b_0 f_n + \cdots + b_k f_{n-k}]$$

where f_n denotes $f(t_n, y_n)$ (for brevity) and a_j, b_j are constants which must be given and determine the specific method.

For explicit linear multistep method, $b_{-1} = 0$.

It is also convenient to define $a_{-1} = 1$, so that the method can we written more concisely as

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f_{n-j} \,.$$

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \dots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}) \,.$$

We consider linear multistep methods with constant step size, which by definition, are methods of the form

$$y_{n+1} = -\mathrm{a}_0 \mathrm{y}_\mathrm{n} - \mathrm{a}_1 \mathrm{y}_{\mathrm{n}-1} - \cdots - a_k y_{n-k} + h[b_{-1} f_{n+1} + b_0 f_n + \cdots + b_k f_{n-k}]$$

where f_n denotes $f(t_n, y_n)$ (for brevity) and a_j, b_j are constants which must be given and determine the specific method.

For explicit linear multistep method, $b_{-1} = 0$.

It is also convenient to define $a_{-1} = 1$, so that the method can we written more concisely as

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f_{n-j} \, .$$

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \ldots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}) \, .$$

Proof.

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \dots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}) \, .$$

Proof.

Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) \; + h b_{-1} f(t_{n+1}, z).$$

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \dots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}).$$

Proof.

Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) + h b_{-1} f(t_{n+1}, z).$$

Now, $F(z)$ is Lipschitz with Lipschitz constant less than or equal to $h|b_{-1}|L$ (why?).

Initial Value Problems: Linear Multistep Methods

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \dots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}).$$

Proof.

Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) + h b_{-1} f(t_{n+1}, z).$$

Now, $F(z)$ is Lipschitz with Lipschitz constant less than or equal to $h|b_{-1}|L$ (why?). By hypothesis ($h < h_0$), the Lipschitz constant is strictly less than 1, that is, $F(z)$ is a contraction.

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \dots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}).$$

Proof.

Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) + h b_{-1} f(t_{n+1}, z).$$

Now, $F(z)$ is Lipschitz with Lipschitz constant less than or equal to $h|b_{-1}|L$ (why?). By hypothesis ($h < h_0$), the Lipschitz constant is strictly less than 1, that is, $F(z)$ is a contraction. The contraction mapping theorem then guarantees a unique fixed point, say y_{n+1}.

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ *where* L *is the Lipschitz constant for* f*. Then for any* $h < h_0$ *and any* $y_n, y_{n-1}, \dots, y_{n-k}$*, there is a unique* y_{n+1} *such that*

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}).$$

Proof.

Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) + h b_{-1} f(t_{n+1}, z).$$

Now, $F(z)$ *is Lipschitz with Lipschitz constant less than or equal to* $h|b_{-1}|L$ *(why?). By hypothesis* $(h < h_0)$*, the Lipschitz constant is strictly less than 1, that is,* $F(z)$ *is a contraction. The contraction mapping theorem then guarantees a unique fixed point, say* y_{n+1}*. Thus, we have* $y_{n+1} = F(y_{n+1})$*, as desired.*

Theorem

Let $h_0 = 1/(|b_{-1}|L)$ where L is the Lipschitz constant for f. Then for any $h < h_0$ and any $y_n, y_{n-1}, \dots, y_{n-k}$, there is a unique y_{n+1} such that

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j}).$$

Proof.

Define

$$F(z) = -\sum_{j=0}^{k} a_j y_{n-j} + h \sum_{j=0}^{k} b_j f(t_{n-j}, y_{n-j}) + h b_{-1} f(t_{n+1}, z).$$

Now, $F(z)$ is Lipschitz with Lipschitz constant less than or equal to $h|b_{-1}|L$ (why?). By hypothesis ($h < h_0$), the Lipschitz constant is strictly less than 1, that is, $F(z)$ is a contraction. The contraction mapping theorem then guarantees a unique fixed point, say y_{n+1}. Thus, we have $y_{n+1} = F(y_{n+1})$, as desired.

Remark

The contraction mapping theorem also implies that the solution can be computed by fixed point iteration as is often done in practice. Moreover, only a fixed (small) number of iterations are made (introducing an additional error).

Module 2

Initial Value Problems

2.4 Implicit method

2.5 Stiffness

2.6 Linear Multistep Methods

- Adams methods

Akash Anand
MATH, IIT KANPUR

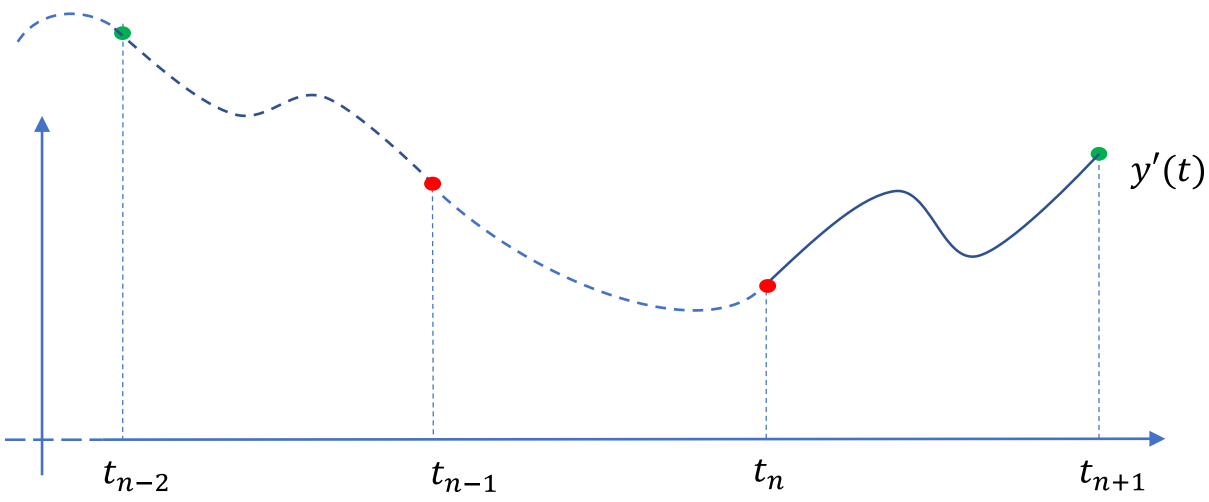
Examples

Adams Bashford methods -

Examples

Adams Bashford methods -

-- 2-step method

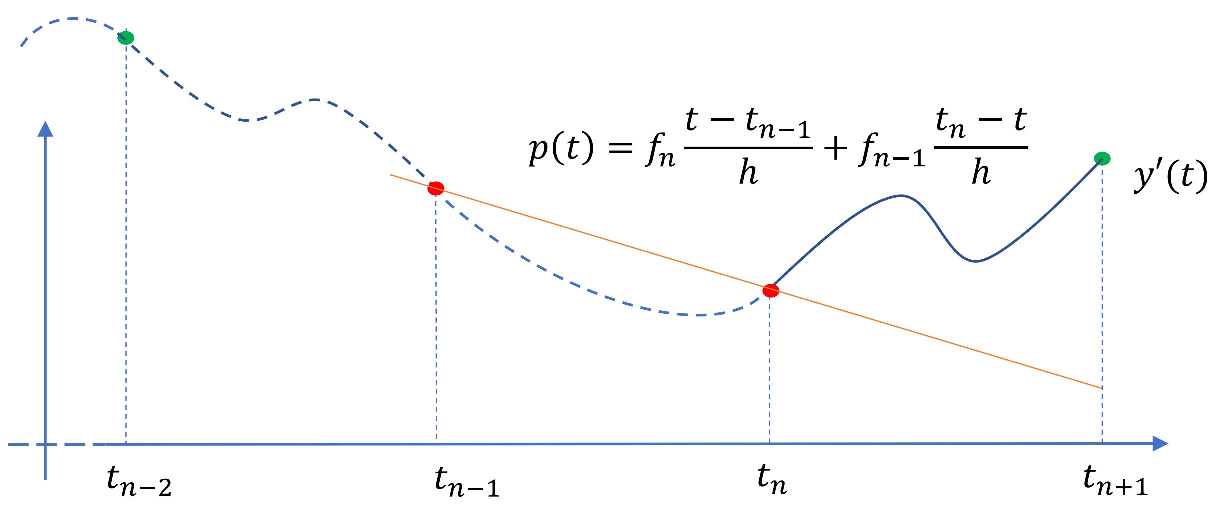


Initial Value Problems: Linear Multistep Methods

Examples

Adams Bashford methods -

-- 2-step method



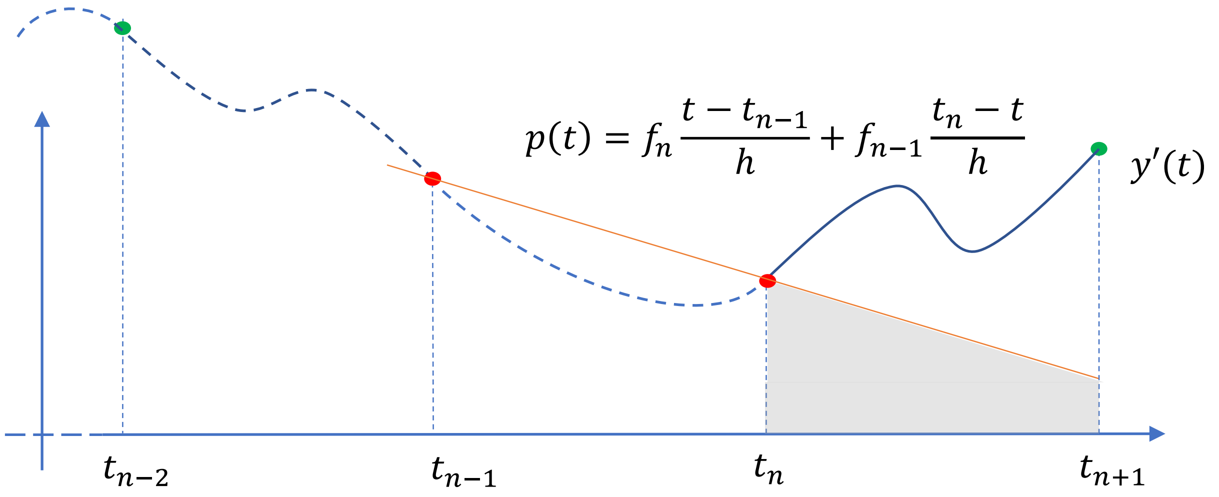
Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Examples

Adams Bashford methods -

-- 2-step method



Initial Value Problems: Linear Multistep Methods

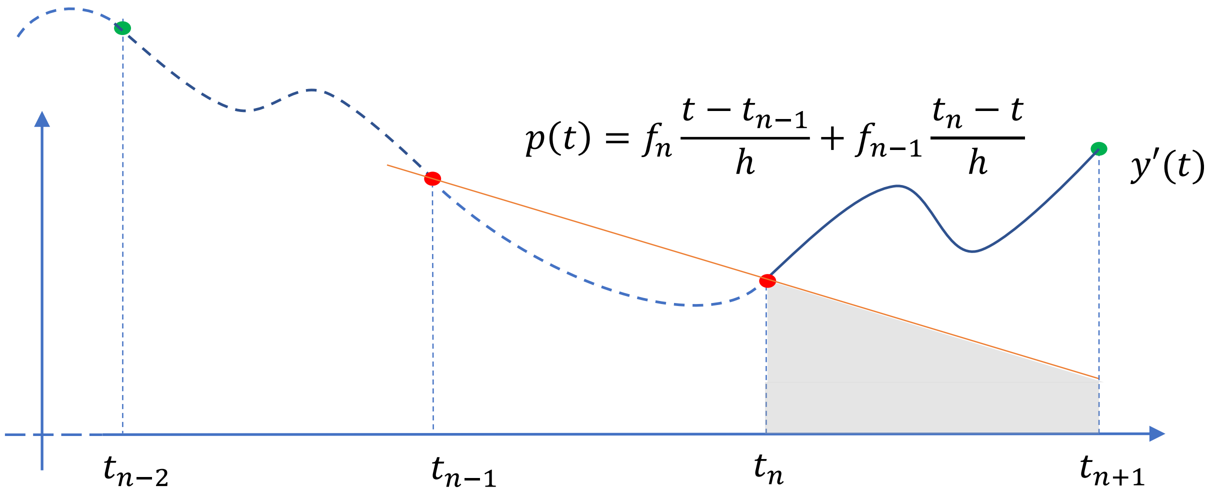
Akash Anand
MATH, IIT KANPUR

Examples

Adams Bashford methods -

-- 2-step method

$$\int_{t_n}^{t_{n+1}} p(t)dt$$



Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

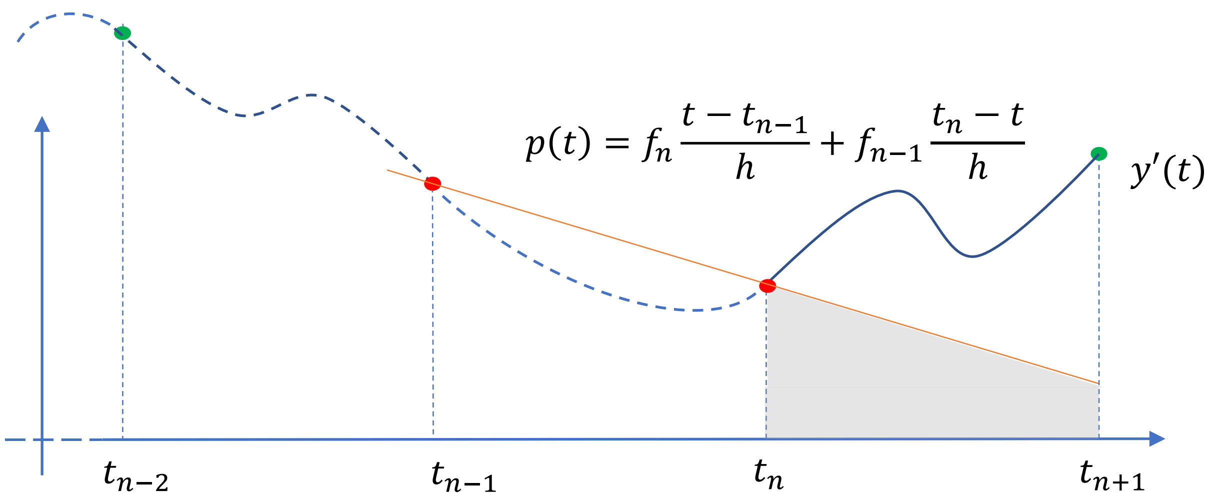
Examples

Adams Bashford methods -

-- 2-step method

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

$$\int_{t_n}^{t_{n+1}} \left(f_n \frac{t-t_{n-1}}{h} + f_{n-1} \frac{t_n - t}{h}\right) dt$$



Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Examples

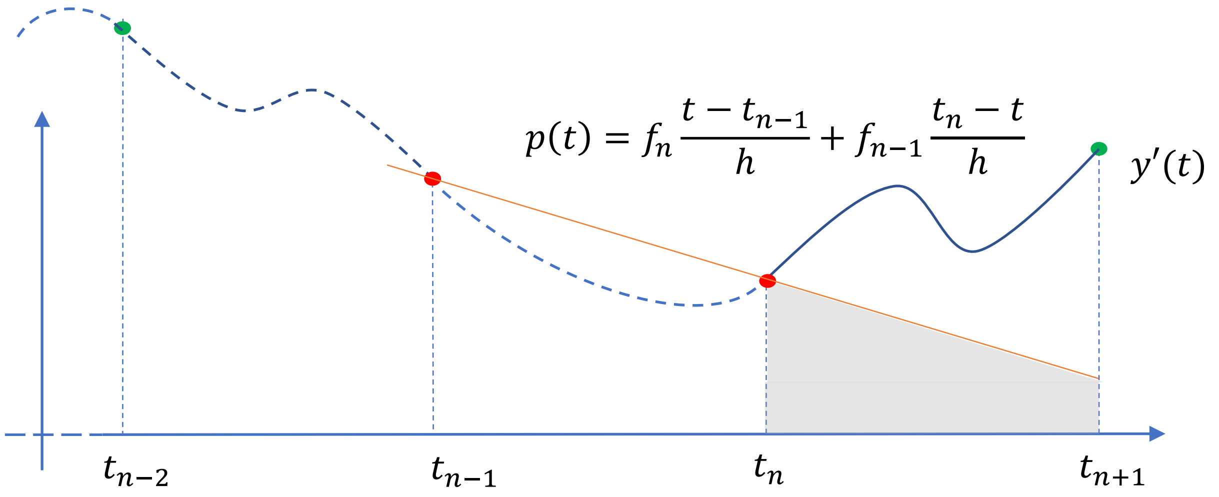
Adams Bashford methods -

-- 2-step method

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

$$\int_{t_n}^{t_{n+1}} \left(f_n \frac{t-t_{n-1}}{h} + f_{n-1} \frac{t_n - t}{h}\right) dt =$$

$$f_n \left(\frac{3h}{2}\right) + f_{n-1} \left(-\frac{h}{2}\right)$$



Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Examples

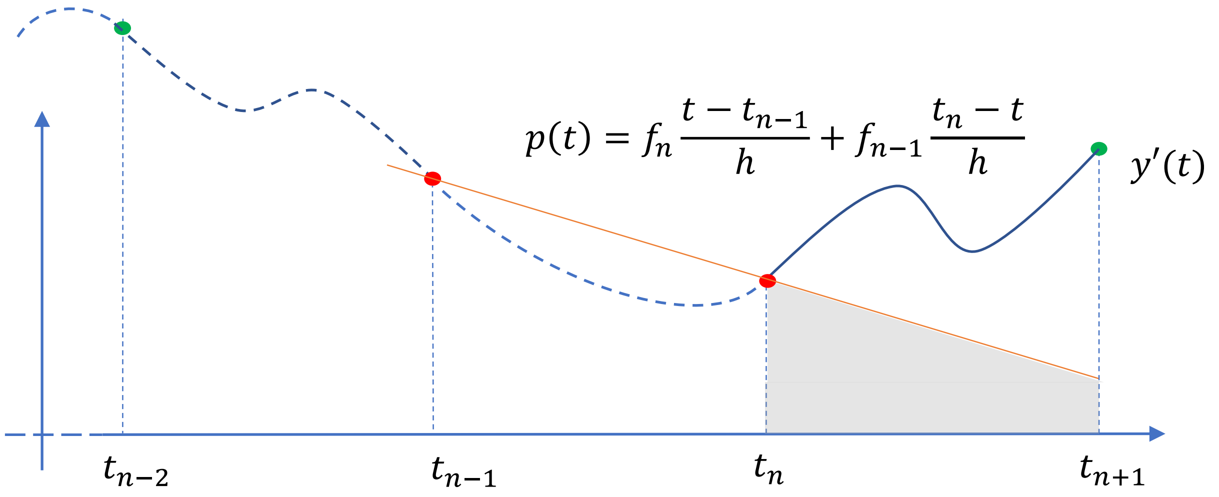
Adams Bashford methods -

-- 2-step method

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

$$\int_{t_n}^{t_{n+1}} \left(f_n \frac{t-t_{n-1}}{h} + f_{n-1}\frac{t_n - t}{h}\right) dt =$$

$$f_n\left(\frac{3h}{2}\right) + f_{n-1}\left(-\frac{h}{2}\right)$$



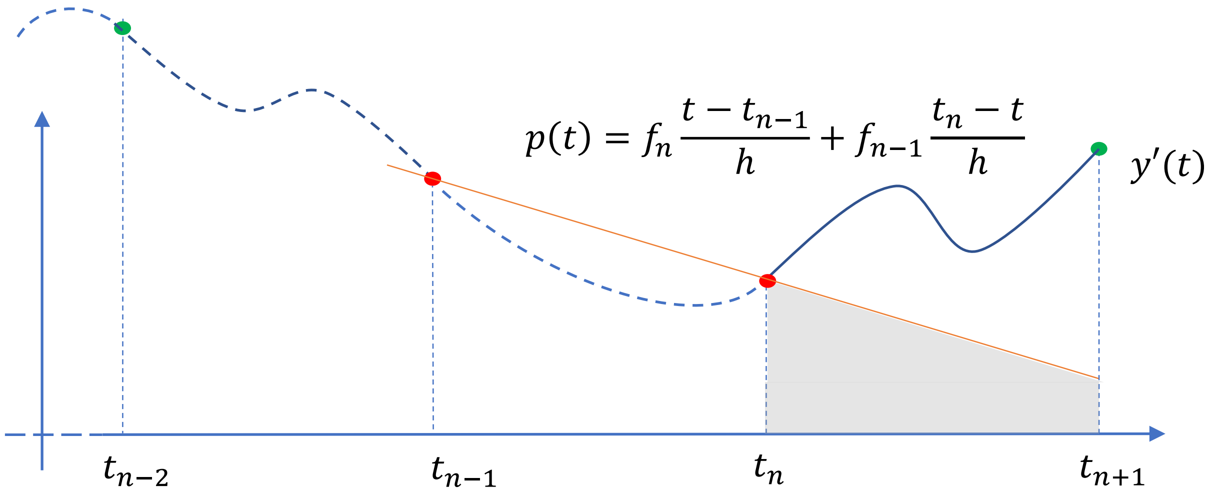
Thus,

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

Examples

Adams Bashford methods -

-- 2-step method



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

$$\int_{t_n}^{t_{n+1}} \left(f_n \frac{t-t_{n-1}}{h} + f_{n-1} \frac{t_n-t}{h}\right) dt =$$

$$f_n \left(\frac{3h}{2}\right) + f_{n-1} \left(-\frac{h}{2}\right)$$

Thus,

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

-- General (k+1) step method

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Examples

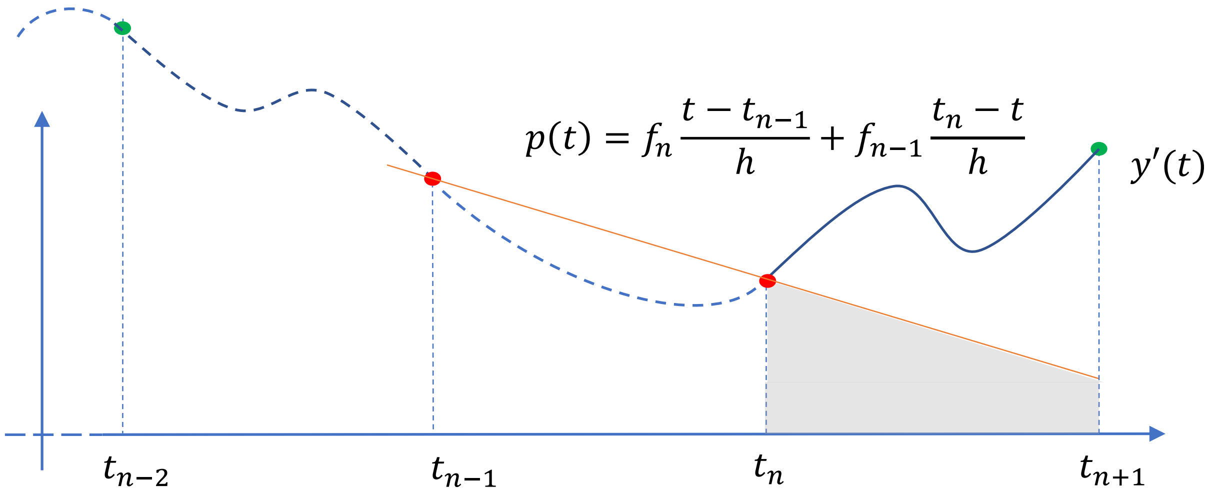
Adams Bashford methods -

-- 2-step method

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

$$\int_{t_n}^{t_{n+1}} \left(f_n \frac{t-t_{n-1}}{h} + f_{n-1} \frac{t_n - t}{h}\right) dt =$$

$$f_n \left(\frac{3h}{2}\right) + f_{n-1} \left(-\frac{h}{2}\right)$$



Thus,

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

-- General (k+1) step method

$$p(t) = \sum_{j=0}^{k} l_j^{(k)}(t) f_{n-j}, \quad \text{where} \quad l_j^{(k)}(t) = \prod_{i=0, i\neq j}^{k} (t - t_{n-i})/(t_{n-j} - t_{n-i})$$

Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Examples

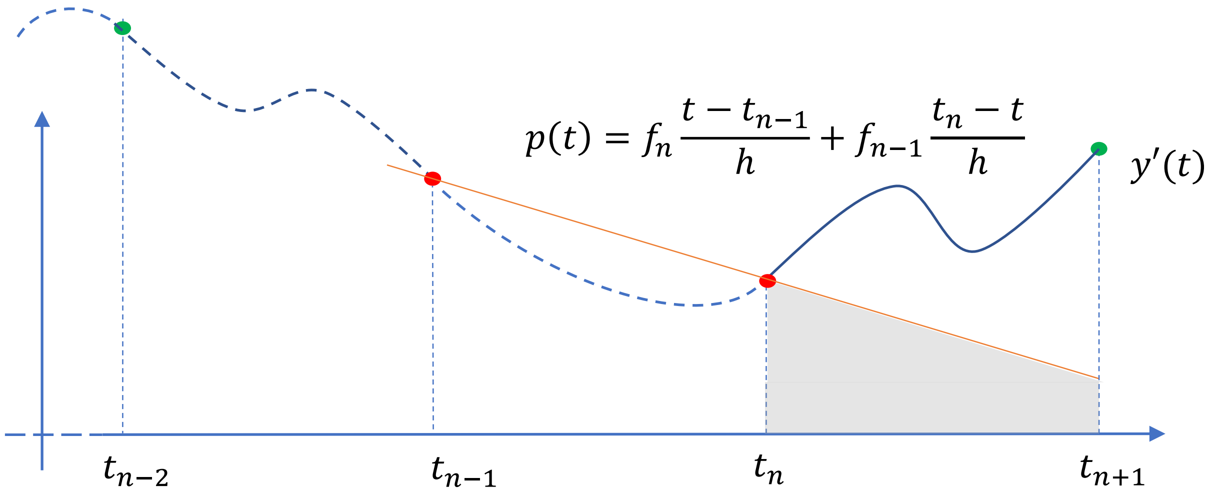
Adams Bashford methods -

-- 2-step method

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

$$\int_{t_n}^{t_{n+1}} \left(f_n \frac{t-t_{n-1}}{h} + f_{n-1} \frac{t_n - t}{h}\right) dt =$$

$$f_n\left(\frac{3h}{2}\right) + f_{n-1}\left(-\frac{h}{2}\right)$$



Thus,

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

-- General (k+1) step method

$$p(t) = \sum_{j=0}^{k} l_j^{(k)}(t) f_{n-j}, \qquad \text{where} \qquad l_j^{(k)}(t) = \prod_{i=0, i\neq j}^{k} (t - t_{n-i})/(t_{n-j} - t_{n-i})$$

Thus,

$$y_{n+1} = y_n + \sum_{j=0}^{k} b_j f_{n-j}, \qquad \text{with} \qquad b_j = \int_{t_n}^{t_{n+1}} l_j^{(k)}(t)dt.$$

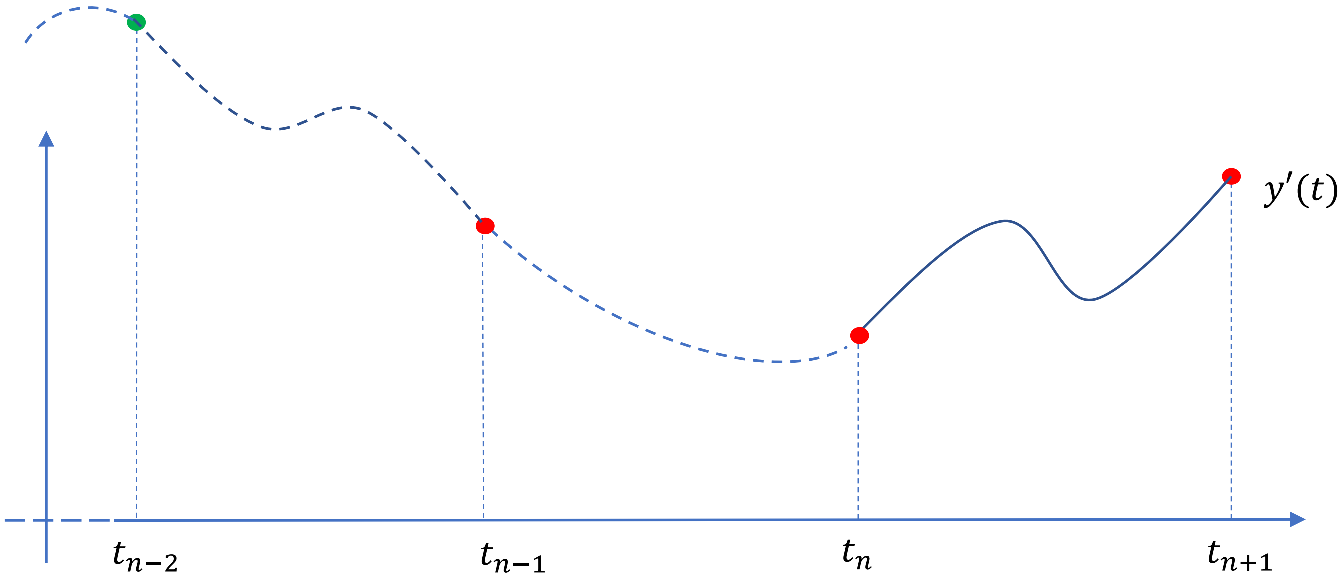
Examples

Adams Moulton methods -

Examples

Adams Moulton methods -

-- 2-step method



Examples

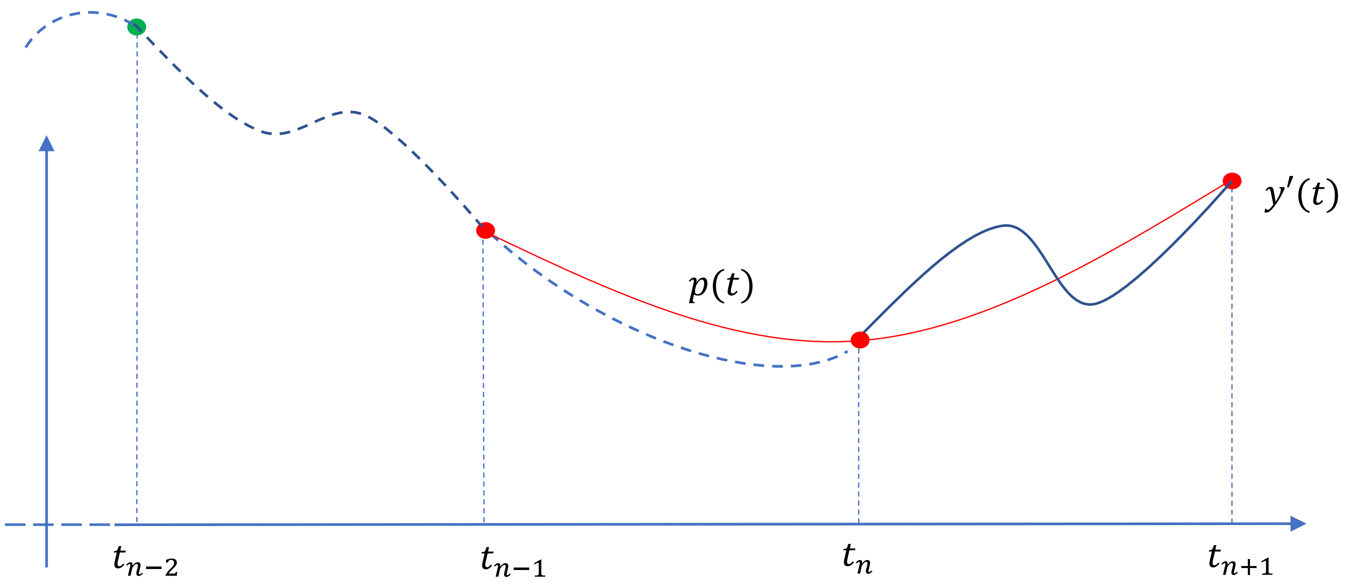
Adams Moulton methods -

-- 2-step method

$$p(t) = f_{n+1}\frac{(t-t_n)(t-t_{n-1})}{2h^2}$$

$$- f_n\frac{(t-t_{n+1})(t-t_{n-1})}{h^2}$$

$$+ f_{n-1}\frac{(t-t_{n+1})(t-t_n)}{2h^2}$$



Initial Value Problems: Linear Multistep Methods

Akash Anand
MATH, IIT KANPUR

Examples

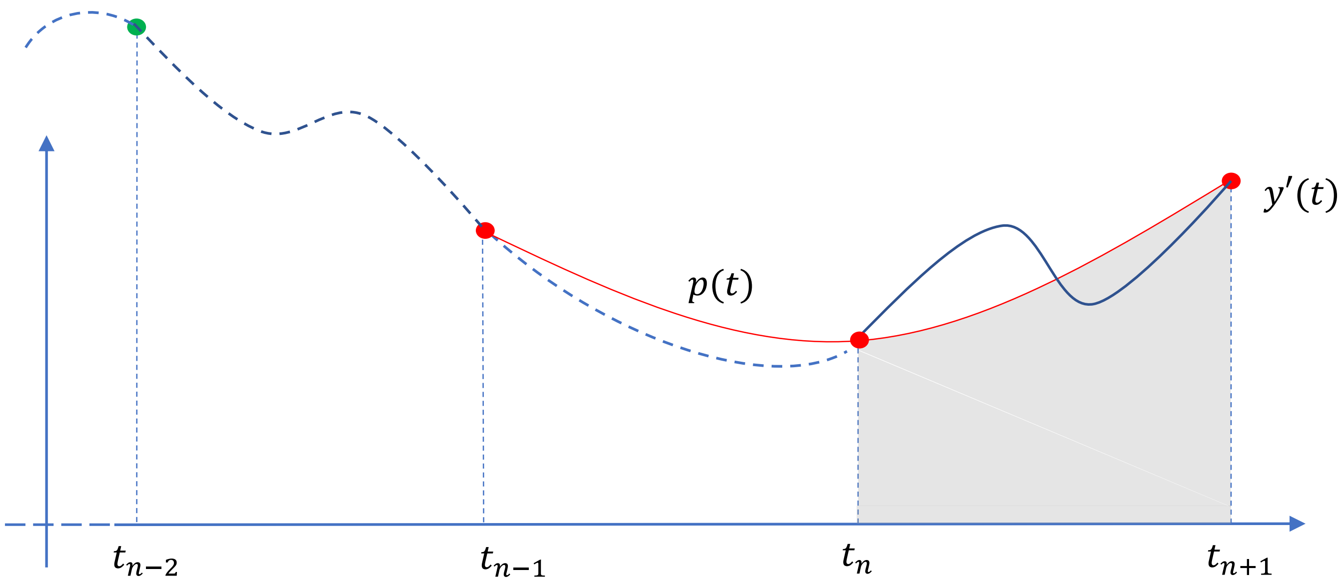
Adams Moulton methods -

-- 2-step method

$$p(t) = f_{n+1}\frac{(t-t_n)(t-t_{n-1})}{2h^2}$$

$$-\, f_n\frac{(t-t_{n+1})(t-t_{n-1})}{h^2}$$

$$+\, f_{n-1}\frac{(t-t_{n+1})(t-t_n)}{2h^2}$$



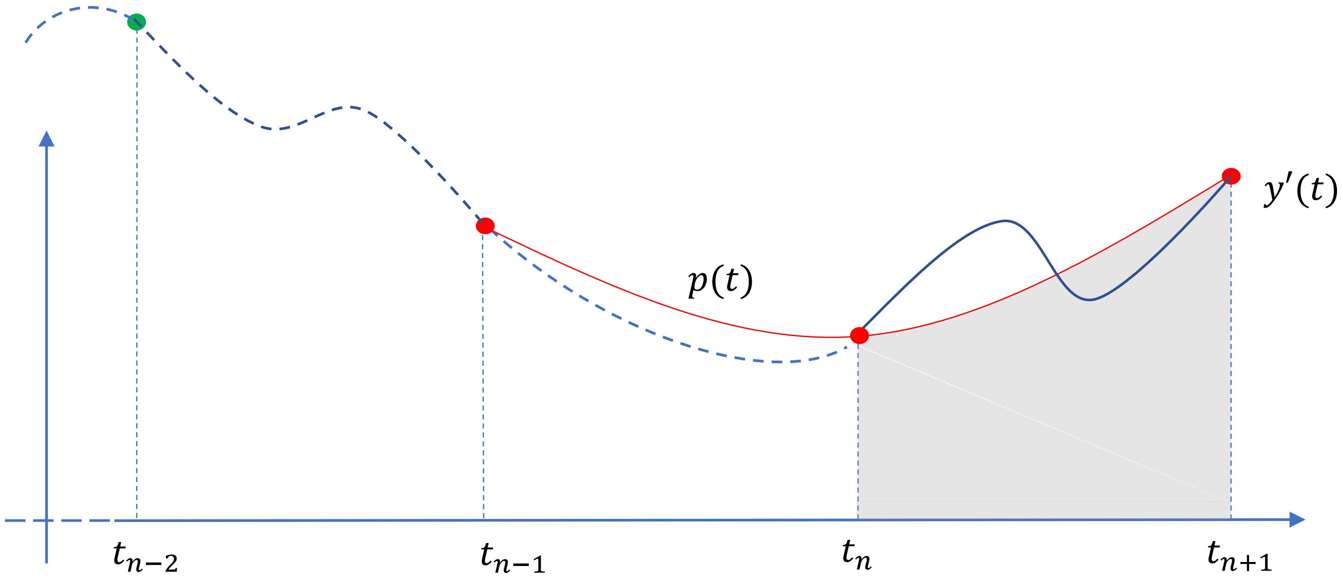
$$\int_{t_n}^{t_{n+1}} p(t)dt$$

Examples

Adams Moulton methods -

-- 2-step method

$$p(t) = f_{n+1}\frac{(t-t_n)(t-t_{n-1})}{2h^2} - f_n\frac{(t-t_{n+1})(t-t_{n-1})}{h^2} + f_{n-1}\frac{(t-t_{n+1})(t-t_n)}{2h^2}$$



$$\int_{t_n}^{t_{n+1}} p(t)dt = f_{n+1}\left(\frac{5h}{12}\right) - f_n\left(-\frac{2h}{3}\right) + f_{n-1}\left(-\frac{h}{12}\right)$$

Examples

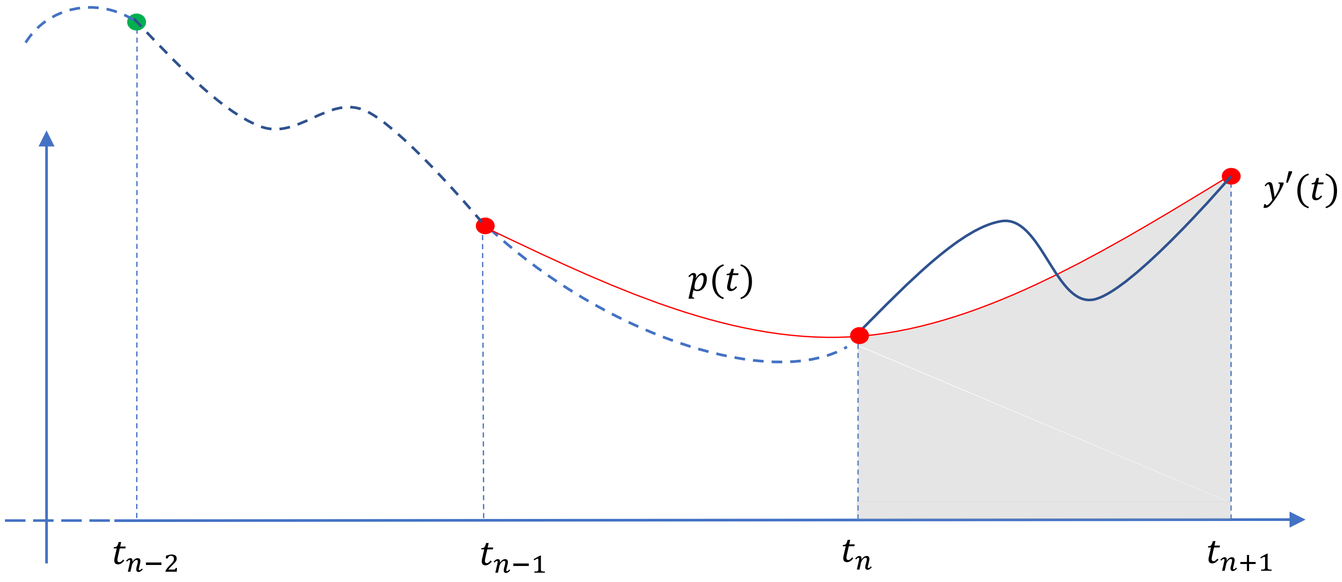
Adams Moulton methods -

-- 2-step method

$$p(t) = f_{n+1}\frac{(t-t_n)(t-t_{n-1})}{2h^2}$$

$$-\, f_n\frac{(t-t_{n+1})(t-t_{n-1})}{h^2}$$

$$+\, f_{n-1}\frac{(t-t_{n+1})(t-t_n)}{2h^2}$$



$$\int_{t_n}^{t_{n+1}} p(t)dt = f_{n+1}\left(\frac{5h}{12}\right) - f_n\left(-\frac{2h}{3}\right) + f_{n-1}\left(-\frac{h}{12}\right)$$

Thus,

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1})$$

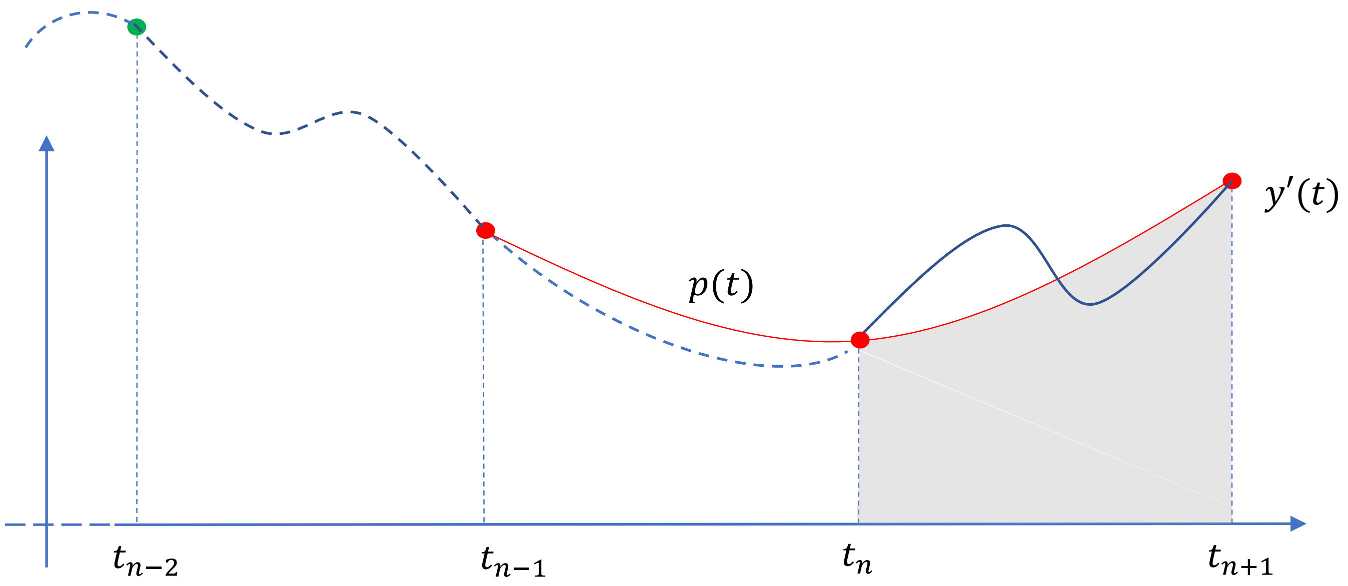
Akash Anand
MATH, IIT KANPUR

Examples

Adams Moulton methods -

-- 2-step method

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1})$$



-- General (k+1)-step method

$$p(t) = \sum_{j=-1}^{k} l_j^{(k)}(t)\, f_{n-j}, \qquad \textit{where} \qquad l_j^{(k)}(t) = \prod_{i=-1, i\neq j}^{k} (t - t_{n-i})/(t_{n-j} - t_{n-i})$$

Thus,

$$y_{n+1} = y_n + \sum_{j=-1}^{k} b_j f_{n-j}, \quad \textit{with} \qquad b_j = \int_{t_n}^{t_{n+1}} l_j^{(k)}(t) dt.$$

Module 2
Initial Value Problems

2.4 Implicit method

2.5 Stiffness

2.6 Linear Multistep Methods

- Consistency and Order

Akash Anand
MATH, IIT KANPUR

Consistency and Order

For the linear multistep method

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j})$$

define the local error as

$$\ell_{n+1}(y,h) = h \sum_{j=-1}^{k} b_j y'(t_n - jh) - \sum_{j=-1}^{k} a_j y(t_n - jh)$$

for any $y \in C^1$, *and* $h > 0$.

Consistency and Order

For the linear multistep method

$$\sum_{j=-1}^{k} a_j y_{n-j} = h \sum_{j=-1}^{k} b_j f(t_{n-j}, y_{n-j})$$

define the local error as

$$\ell_{n+1}(y,h) = h \sum_{j=-1}^{k} b_j y'(t_n - jh) - \sum_{j=-1}^{k} a_j y(t_n - jh)$$

for any $y \in C^1$, and $h > 0$.

The linear multistep method is consistent if

$$\lim_{h\to 0} \max_{k \le n < N} \left| \frac{\ell_{n+1}(y,h)}{h} \right| = 0$$

for all $y \in C^1$.