# Math 385 Notes

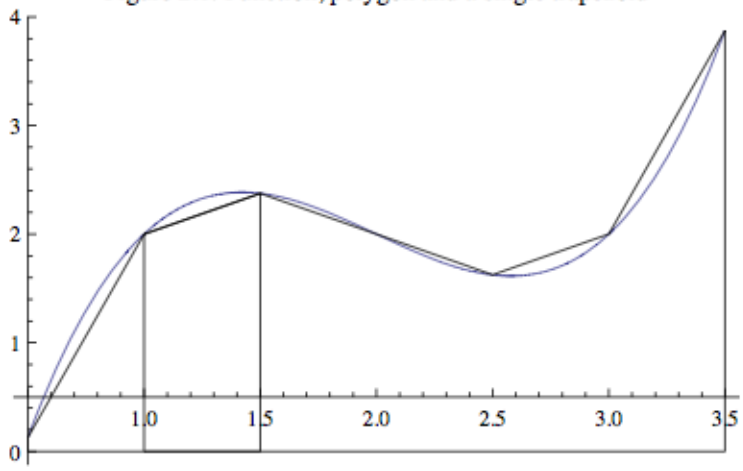
Mario L. Gutierrez Abed

## Numerical Integration Techniques

### TRAPEZOID METHOD

Let $f$ be a bounded real valued function on $[a, b]$ and suppose we partition the interval $a = x_0 < x_1 < \ldots < x_n = b$. If we know $f$ or at least half the values $f(x_i)$, then we can easily approximate the integral of $f$ over $[a, b]$. Indeed, if $f$ is strictly positive on $[a, b]$ and we join the points $(x_i, f(x_i))$ with line segments, then the area under the resulting polygon will approximate the area under $f$. Since the figure formed by connecting the four points $(x_{i-1}, 0)$, $(x_i, 0)$, $(x_i, f(x_i))$ and $(x_{i-1}, f(x_{i-1}))$ is a trapezoid, we will get the area under the polygon as the sum of areas of trapezoids. ( see figure below)

Figure 2.1: Function, polygon and a single trapezoid




Using the standard formula for the area of a trapezoid (length of base × average of the heights), we have

(I) $$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} \frac{f(x_i)+f(x_{i-1})}{2}\,(x_i - x_{i-1})$$ (Trapezoid Method)

This approximation technique is called the trapezoid method. It seems to be intuitively obvious, easy to implement, and hence a reasonable procedure. However, later we will learn that it is not as accurate as other less intuitive processes.

**Note:** In previous sections we have developed procedures to approximate an unknown by parametric cubics. For these cases the implementation of the trapezoid method or any of the techniques developed below is considerably more difficult.

Example:

Take a partition $a = \gamma(t_0) < ... < \gamma(t_n) = b$, where $\gamma(t) = (x(t), y(t))$. For each $i$, we must locate $t_i$ by solving $x(t) = x_i$ using Newton's method or similar. Once we do that, we can then evaluate $y(t_i)$ to get the height at $x_i$. Hence equation (I) becomes

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} \frac{y_i + y_{i-1}}{2} (x_i - x_{i-1})$$

where $x(t_i) = x_i$ and $y(t_i) = y_i$. Now we need to locate the $t_i$. For this purpose, as mentioned above, we can use Newton's method via the FindRoot function.

```
FindRoot[x[t] == x_i, {t, τ_i}];
```

where $\tau_i$ is the estimate for $t_i$ required by the method.
We claim that $\tau_i = t_{i-1}$ is a reasonable estimate. First of all we know $t_0$. Second, in order for the trapezoid method to produce a reasonable estimate of the integral, the $x_i$ must be close together. But if $x$ is continuous or nearly continuous, then the $t_i$ will also be close.
Nevertheless, Newton's method must always be suspect as it may produce a $t_i$ which is not between $a = t_0$ and $b = t_n$. Therefore, it is best to test the value returned for reasonableness. In particular, we expect $t_i$ to be between $a$ and $b$. Further it is reasonable to expect that $t$ increases as $x$ increases. In other words, we need to ensure that $a < t_i < b$ and $t_i > t_{i-1}$.
We may implement this in the following manner :

```
 tVal = FindRoot[x[t] == x_i, {t, τ_i}];
If[tVal[[1, 2]] < τ_1 ⋁ tVal[[1, 2]] > t_n,
     Print["tVal is out of bounds"];
     Break[ ],
      ** Continue processing **
  ];
```

One important application of numerical integration is computing arc length.

If $f(x) = y$ with $x \in [a, b]$, the length of $f$ is given by

$$\text{length} = \int_a^b \sqrt{1 + [f'(x)]^2}\, dx.$$

If the curve is instead given parametrically with $\gamma(t) = (\gamma_1(t), \gamma_2(t))$ and $t \in [\alpha, \beta]$, then we have

$$\text{length} = \int_\alpha^\beta \sqrt{[\gamma_1'(t)]^2 + [\gamma_2'(t)]^2}\, dt$$

## SIMPSON' S RULE

Now let's look at an alternative to the trapezoid method. Notice that the area of the trapezoid returns the actual value of the integral when the function is linear. But most functions that we consider have curved graphs, not polygonal graphics. Simpson's Rule is characterized by the fact that it returns the actual integral for certain quadratics. In other words, it assumes that the graph of the function is more or less a parabolic.

First we state Simpson's rule and then we will verify the claim.
** A neat derivation of Simpson's rule can be found on Stewart's Single Variable Calculus pg 511-513 **

For an interval $[a - h, a + h]$, the estimated integral via Simpson's rule is

$$\int_{a-h}^{a+h} f(x)\, dx \approx 2h\left(\frac{1}{6} f(a-h) + \frac{2}{3} f(a) + \frac{1}{6} f(a+h)\right)$$

If we have a parabolic function, say $f(x) = (x-a)^2$, then

$$\begin{aligned} 2h\left(\frac{1}{6} f(a+h) + \frac{2}{3} f(a) + \frac{1}{6} f(a-h)\right) &= 2h\left(\frac{1}{6}(a+h-a)^2 + \frac{2}{3}(a-a)^2 + \frac{1}{6}(a-h-a)^2\right) \\ &= 2h\left(\frac{1}{6} h^2 + \frac{1}{6} h^2\right) = \frac{2}{3} h^3 \end{aligned}$$

which is exactly the integral of $f$. Therefore, Simpson's Rule is exact in the case of the quadratic polynomial centered at the interval midpoint. Now we recast the technique applied to a larger interval by summing all the areas. Suppose we have an interval $[a, b]$ together with a partition, $a = x_0 < x_1 < \ldots < x_n = b$. Then Simpson's rule states that

(II) $$\int_a^b f(x)\, dx \approx \sum_{i=1}^{n} (x_i - x_{i-1})\left(\frac{1}{6} f(x_{i-1}) + \frac{2}{3} f\left(\frac{x_i + x_{i-1}}{2}\right) + \frac{1}{6} f(x_i)\right)$$ (Simpson's Rule)

## MIDPOINT METHOD

The midpoint method is remarkable because of both its simplicity and its accuracy. As before we begin with a function $f$ defined on some interval $[a, b]$ and then we partition the interval $a = x_0 < x_1 < \ldots < x_n = b$. For each of the subintervals $[x_{i-1}, x_i]$, let $\alpha$ denote the midpoint and let $2h$ be the length of the subinterval. The subinterval now becomes $[\alpha - h, \alpha + h]$ and the trapezoid approximation for the integral is

$$2h\left(\frac{f(\alpha-h)+f(\alpha+h)}{2}\right) = h\,(f(\alpha-h)+f(\alpha+h))$$

If we replace $f(\alpha-h)+f(\alpha+h)$ with $2f(\alpha)$ then we get the midpoint rule. In this case equation (I) becomes

(III) $$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} 2h\,f(\alpha) = \sum_{i=1}^{n} (x_i - x_{i-1})\,f\left(\frac{x_i+x_{i-1}}{2}\right)$$ (Midpoint Method)

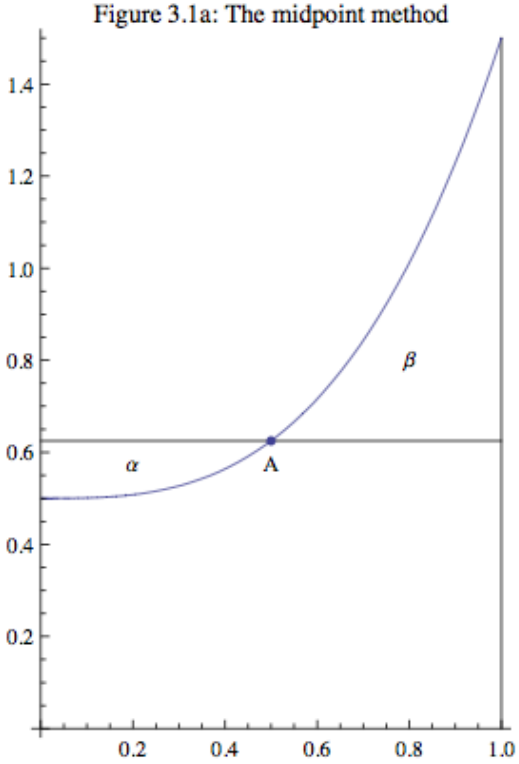
## ERRORS FOR MIDPOINT AND TRAPEZOID METHOD

Now suppose that $f$ is strictly concave up or concave down on $[a, b]$. The error for the midpoint method is given by the area between the rectangle generated by the method and the function $f$. If the function is strictly concave up or concave down in the interval $[x_{i-1}, x_i]$, then the midpoint method is better than the trapezoid method. More precisely, the absolute error from the midpoint method is smaller than the absolute error from the trapezoid method. The proof is a simple argument using elementary geometry.

**Take-away Note:** Midpoint Error < Trapezoid Error so long as there is not an inflection point of $f$ on $[a, b]$. In the case that $f$ is linear, the Trapezoid and Midpoint methods are exact.

Suppose that the function is concave up in the interval and then consider Figure 3.1a below. The error for the midpoint rule is the area between the horizontal line and the curve. We say that the area to the left of the point $A$, denoted by $\alpha$, is positive and the area on the right ($\beta$) is negative **this is arbitrary ,i.e. we could've chosen $\alpha$ to be negative and $\beta$ to be positive**. Denoting the error to the right as $-\beta$, we have that the midpoint method error is $E_M = \alpha - \beta$.

Figure 3.1a: The midpoint method




If we then include the tangent line to $f$ at the midpoint (see figure 3.1b below), then triangles $\triangle$ ADE and $\triangle$ABC are congruent (by side-angle-side).
Now let $\zeta$ be the common area of the two triangles, then we have that $\beta = \delta + \zeta$ and $\alpha = \zeta - \gamma$.
Hence we may rewrite the midpoint error as

$$E_M = \alpha - \beta = \zeta - \gamma - (\delta + \zeta) = -(\gamma + \delta) \qquad \text{(Midpoint Method Error)}$$

**see figure below to see what all these letters represent**.
This last term is in fact the error if we had used the tangent line to compute the approximate integral. Taking absolute values, the absolute midpoint method error is $|E_M| = \gamma + \delta$.

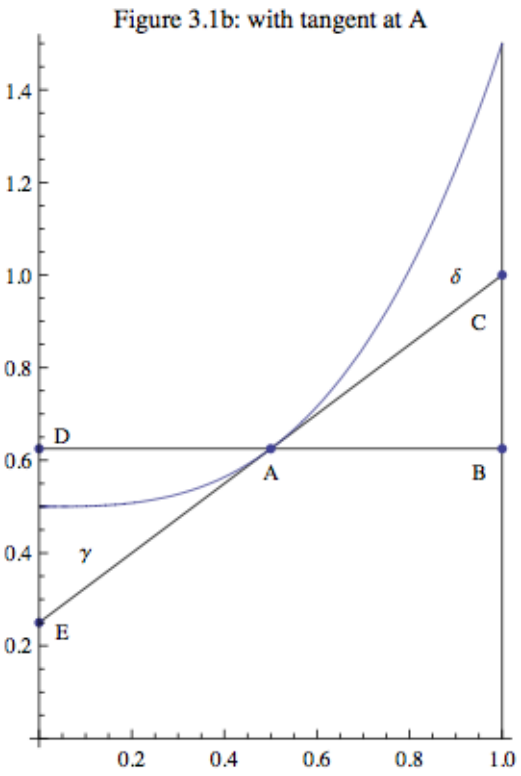
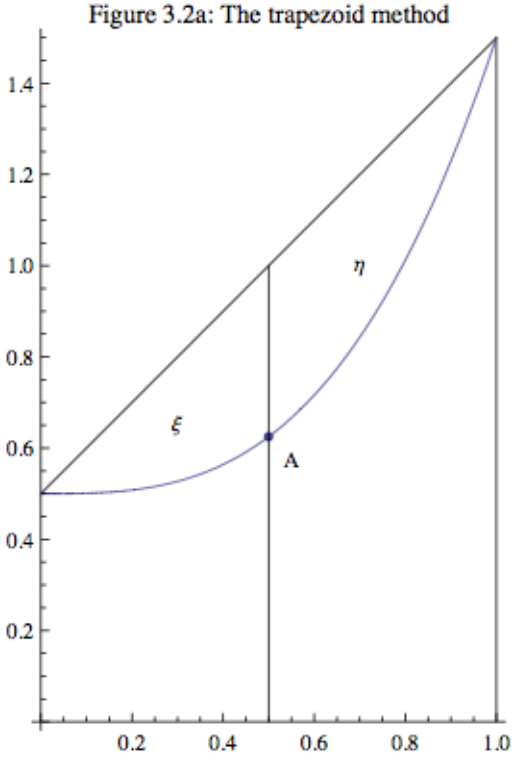



Figure 3.2a below shows the trapezoid method error:

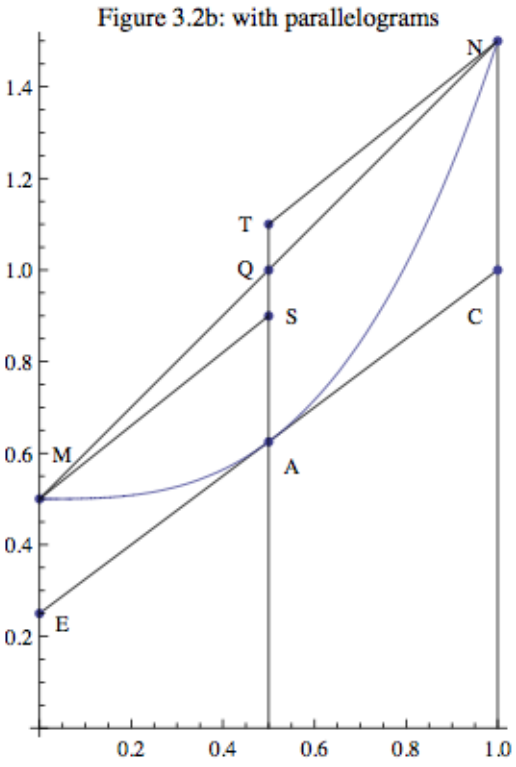
$E_T = \xi + \eta$ (Trapezoid Method Error)

Figure 3.2a: The trapezoid method




Next we construct line segments from $M$ and $N$ parallel to the tangent (see figure 3.2b below). We now have two parallelograms $P_1$ : EASM and $P_2$ : ACNT. Since the triangles △MSQ and △QNT are congruent (by angle-side-angle), then $E_T$ is equal to the area of $P_1$ and $P_2$ that lies above the curve.

On the other hand the absolute midpoint method error is equal to the portion of the area of the two parallelograms that lies below the curve. Since the curve is concave up on $[a, b]$, then necessarily $E_T > E_M$, i.e. the error of the trapezoid method is greater than the error of the midpoint method. This implies that the midpoint method is more accurate than the trapezoid method when $f$ is strictly concave up or down.




## GAUSSIAN QUADRATURE

While the *Mathematica* function Integrate[ ] computes symbolic integrals (via an antiderivative if one exists), the function NIntegrate[ ] performs numerical integration using a process called Gaussian quadrature. Alternatively, if while using Integrate[ ], *Mathematica* fails to find an antiderivative or the function is one of the functions known to not have an anti-derivative, then *Mathematica* will automatically execute Gaussian quadrature, i.e. it automatically switches to NIntegrate[ ].

There are many different realizations of quadrature. In its simplest form (Gaussian quadrature), the numerical integration is

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$ (Gaussian Quadrature Formula)

where the quadrature points $x_i$ and weights $w_i$ depend on $n$.

On *Mathematica* we use the GaussianQuadratureWeights[$n$, $a$, $b$] function, which returns a list of the $n$ pairs $\{x_i, w_i\}$ of the elementary $n$-point Gaussian formula for quadrature on the interval $a$ to $b$, where $w_i$ is the weight of the abscissa $x_i$.

We can also add an extra argument to this function, GaussianQuadratureWeights[$n$, $a$, $b$, $prec$ ] , in an attempt to give a result with prec digits of precision.
This and other functions can be accessed through loading a library of special analysis functions by executing the following line:
<< NumericalDifferentialEquationAnalysis` .

Some properties of Gaussian quadrature:
Given a function $f$ with at least $2n+1$ continuous derivatives on $[a, b]$, then the $n+1$ Gaussian points and weights are determined so that

i) There's a polynomial $p$ that interpolates $f$ at the $n+1$ points.
ii) The degree of $p$ is $2n$.
iii) $f(x) - p(x) = \frac{f^{(2n+1)}}{(2n+1)!} \prod_{i}^{n+1} (x - x_i)$ .
iv) $\int_a^b p(x)\, dx = \sum_{i=1}^{n+1} f(x_i)\, w_i$ .

One point quadrature (midpoint method) is commonly used to compute the integrals that occur in finite element method. Two point quadrature is used in collocation. Both are techniques for computing approximate solutions of differential equations.

**Note:** The midpoint method is a special case of Gaussian quadrature, i.e. it is a one-point Gaussian quadrature. We have that the following is true:
- One-point GQ is exact for functions of degree 1.
- Two-point GQ is exact for functions of degree ≤ 3.
- Three-point GQ is exact for functions of degree ≤ 4.

## AN APPLICATION : FIRST ORDER ODE' S

We consider the following first order ordinary differential equation,
(IV) $\frac{d}{dx}(u) = f(x, u)$
where $u$ is a function defined and differentiable on an interval $[a, b]$, and $f$ is a function both of the independent variable $x$ and the dependent variable $u$. Our purpose here is to illustrate some of the techniques developed above. Throughout this section we will suppose that the boundary value, $u(a) = u_0$ is known.

In the simplest cases, a diff equation in the form of (IV) is just a straightforward separable equation. However, if this is not possible there are a host of numerical techniques at our disposal. We will

spend the remainder of this section developing three of these techniques, namely the Forward Euler, Forward Euler with corrector, and the midpoint method.

We begin by setting up the usual partition on $[a, b]$, i.e. $a = x_0 < \ldots < x_n = b$. By assumption we know $u(x_0)$. Our goal then is to estimate $u(x_i)$, with $i > 0$. In each case the process will be iterative. We will use knowledge of $u(x_0)$ to estimate $u(x_1)$ and then derive the estimate of $u(x_2)$ from the prior estimate of $u(x_1)$ and so forth. The obvious flaw is that any error introduced at the $i^{\text{th}}$ iteration is likely compounded at the next.

▸ Forward Euler:

The simplest technique is called forward Euler.
In this case we first formally integrate (IV) to get

(V) $u(x_{i+1}) - u(x_i) = \int_{x_i}^{x_{i+1}} f(x, u(x))\, dx$

The question is how to estimate the integral on the right hand side of (V). At the $i + 1$ step in the iteration, we know $u(x_i)$, and hence we know $f(x_i, u(x_i))$. In forward Euler, we replace the integral with $f(x_i, u(x_i))\,(x_{i+1} - x_i)$ and set

(VI) $u(x_{i+1}) = u(x_i) + f(x_i, u(x_i))\,(x_{i+1} - x_i)$ (Forward Euler)

where $f(x_i, u(x_i)) = u'(x_i)$ and $x_{i+1} - x_i = \Delta x$.

Example:

Take $u(x) = 1 - e^{-\frac{x}{5}}$. Thus $u'(x) = \frac{1}{5} e^{-\frac{x}{5}} = \frac{1}{5}(1 - u)$. We set the initial condition $u(0) = 0$ and take $\Delta x = 1$.
Then,

$u_0 = u(0) = 0$

$u_1 = u_0 + u_0' \Delta x$
$= 0 + \frac{1}{5}(1 - 0) \cdot 1$
$= \frac{1}{5} = 0.2$

$u_2 = u_1 + u_1' \Delta x$
$= 0.2 + \frac{1}{5}(1 - 0.2)$
$= 0.2 + 0.16 = 0.36$
and so forth.....

▸ Forward Euler w/ corrector:

Example:

Again, we use the same curve $u(x) = 1 - e^{-\frac{x}{5}}$, with $u'(x) = \frac{1}{5}(1-u)$, and we set $\Delta x = 1$ and the initial condition

$u_0 = u(0) = 0.$

Then we get $u_1$ by using Forward Euler as above,

$u_1 = u_0 + u_0' \Delta x$

This time we take the derivative at $u_1$,

$u_1' = \frac{1}{5}(1 - u_1)$

and then use the corrected derivative at $u_1$, which is given by

$\hat{u}_1' \approx \frac{1}{2}(u_1' + u_0') \approx \frac{1}{2}\left(f(x_1, u_1) + f(x_0, u_0) \approx \frac{1}{10}((1-u_1) + (1-u_0))\right.$

Then we use $\hat{u}_1'$ to calculate $u_2$:

$u_2 = u_1 + \hat{u}_1' \Delta x$

and so forth.....

In general we have

$$\hat{u}_{i+1}' \approx \frac{1}{2}(u_{i+1}' + u_i') \approx \frac{1}{2}\left(f(x_{i+1}, u_{i+1}) + f(x_i, u_i) \approx \frac{1}{10}((1-u_{i+1}) + (1-u_i))\right.$$

Thus, the corrected Forward Euler method is given by

(VII) $u_{i+1} = u_i + \hat{u}_{i+1}'(x_{i+1} - x_i)$ (Forward Euler corrected)

Certainly the error introduced at each iteration of **(VI)** is a function of $(x_{i+1} - x_i)$. There is another problem with **(VI)**. The left hand end point is not a particularly good estimator of the integral. The midpoint would be better. The following technique, due to Runge, implements improvements in both directions.

▸ Midpoint/Runge method:

We begin by setting $\xi_i = \frac{1}{2}(x_{i+1} + x_i)$, which is the midpoint for each subinterval $[x_i, x_{i+1}]$. We use (VI) to estimate $u(\xi_i) = u(x_i) + \frac{1}{2} f(x_i, u(x_i))(x_{i+1} + x_i)$, as if we were applying Euler's method at the midpoint. Then we estimate the slope as $f(\xi_i, u(\xi_i))$ and thus we get

(VIII) $u(x_{i+1}) = u(x_i) + f(\xi_i, u(\xi_i))(x_{i+1} - x_i)$ (Midpoint/Runge method)

$u(\xi_1) = u_0 + \frac{1}{2} f(x_0, u_0) \Delta x$

Apply the ODE:

$u'(\xi_1) = f(\xi_1, u(\xi_1))$.

Then set $u_1 = u_0 + u'(\xi_0) \Delta x$

and so forth.....

▸ Taylor method:

Notice that the iterative process of the midpoint/Runge method looks like a truncated Taylor expansion (a Taylor expansion of order 1). So let's consider making this method more accurate by adding further terms in Taylor expansion. But for this to work we need $u(x)$ to have a second derivative that exists.

Then we have

(IX) $u(x_{i+1}) = u(x_i) + f(x_i, u(x_i))\,(x_{i+1} - x_i) + \frac{1}{2} f'(x_i, u(x_i))\,(x_{i+1} - x_i)^2$ (Taylor order 2)