Integration

1 Preamble

In engineering analysis we must often calculate integrals of univariate functions; for example, in order to predict the hydrostatic pressure in the ocean — to be withstood by a submersible — we must integrate the specific weight with respect to depth. Unfortunately, in most cases we are not provided with a closed-form representation of the function, and instead we must base our prediction on limited (experimental) observations or (computational) evaluations. In this nutshell we answer the following question: if we can probe a function at some finite number of input values, how we can estimate the integral of the function over an interval? We consider here one approach to this approximation problem: integration through interpolation.

We introduce in this nutshell the general "integration through interpolation" framework, and we present particular integration schemes, also known as "quadrature" rules : rectangle, left; rectangle, right; rectangle, middle (or midpoint); and trapezoid. For each scheme we develop error bounds based on interpolation error estimates, we provide an analysis of the convergence rate, and we summarize the associated operation counts in FLOPs. We also raise the issues of resolution and smoothness within the context of numerical integration.

Prerequisites: univariate differential and integral calculus; basic principles of numerical methods: discretization, convergence, convergence order, refinement, resolution, operation counts and FLOPs, asymptotic and big-O estimates; Interpolation: formulation, error analysis, and computational considerations.

2 Motivation: Example

Let us consider a concrete example of numerical integration. We wish to estimate the hydrostatic pressure in the ocean at some depth of interest; recall that the hydrostatic pressure at depth of interest d is given by

$$p(d) = \int_0^d \rho(x) g dx + p_{\text{atm}},$$

where $\rho(x)$ is the density of the water at depth x, g is the gravitational acceleration, and p_{atm} is the atmospheric pressure. (Note that depth shall always refer to vertical distance below the free surface.) The gravitational acceleration can be safely assumed constant, so we effectively need to evaluate the integral

$$I(d) = \int_0^d \gamma(x)dx,$$

for $\gamma(x) \equiv \rho(x)g$ the specific weight at depth x. Note that x is the "dummy" variable of integration and d is the depth of interest: the integral depends on the parameter d; for the most part, as regards the focus of this nutshell, we may think of d as some fixed depth of interest.

To accomplish our task, we are given Table 1, which provides the specific weight of ocean water at seven different depths. (The density, and hence the specific gravity, of ocean water depends on the salinity and temperature, which in turn varies with depth.) How can we exploit the data of Table 1 to approximate the hydrostatic pressure at a depth of interest $d \equiv 600$m? Can we say anything about the accuracy of this estimate? What is the cost associated with the computation of this pressure estimate? The material provided in this nutshell will help you answer these questions, not just for our particular example, but also more generally for any univariate integration.

depth (m)	0	100	200	300	400	500	600
specific weight (kPa/m)	10.055	10.059	10.063	10.068	10.072	10.076	10.079

Table 1: Variation in the specific weight of ocean water with depth at a particular point.[1]

3 Rectangle, Left Rule

We are often given some function of a single variable, x, say $f(x)$, which we wish to integrate from $x = a$ to $x = b$. We express the desired integral as

$$I = \int_a^b f(x)\,dx\ .$$

Geometrically, we may interpret the integral as an area, as shown in Figure 1.

To perform this integral we introduce a discretization: we shall break the interval (a, b) into $N - 1$ segments each of length $h = (b - a)/N$. To be a bit more precise, we introduce the set of segment endpoints $a \equiv x_1 < x_2 < \cdots < x_N \equiv b$, from which we may then define our segments as $S_1 \equiv (x_1, x_2), S_2 \equiv (x_2, x_3), \ldots, S_{N-1} \equiv (x_{N-1}, x_N)$. The segment endpoints and segments are depicted in Figure 2. In what follows, we shall assume that all segments are of the same length, though in all cases the techniques described readily extend to the case of non-uniform grids.

We next express the area as a sum of areas associated with each of the segments,

$$I = \int_a^b f(x)\,dx = \int_{S_1} f(x)\,dx + \int_{S_2} f(x)\,dx + \ldots + \int_{S_{N-1}} f(x)\,dx = \sum_{i=1}^{N-1} \int_{S_i} f(x)\,dx\,,$$

[1] This table is derived from the representative ocean density profile provided in *Windows to the Universe*.

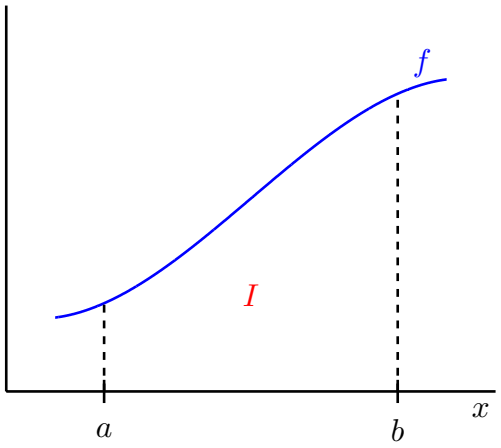

Figure 1: Illustration of the integration process.

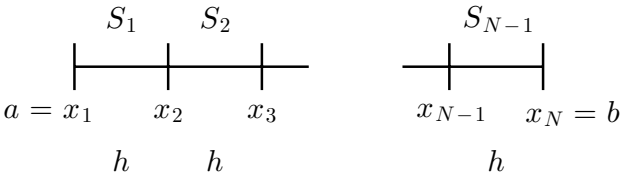

Figure 2: Discretization of the segment (a, b) into $N - 1$ segments using N data points.

where

$$\int_{S_1} f(x)\,dx \equiv \int_{x_1}^{x_2} f(x)\,dx\,,$$

and more generally, for $1 \le i \le N-1$,

$$\int_{S_i} f(x)\,dx \equiv \int_{x_i}^{x_{i+1}} f(x)\,dx.$$

We note that technically in each segment we may consider the limit of f from within the segment, in that way gracefully treating discontinuities. (It is for this reason that we choose here to define our segments as open intervals.)

CYAWTP 1. Consider the uniform discretization of Figure 2 for $N = 3$ and hence $N - 1 = 2$ segments. Sketch on Figure 1 the area which corresponds to the term $\int_{S_2} f(x)\,dx$.

We now consider the integral over one segment, say S_1. In general, we will not be able to perform the requisite integrations over each segment in closed form. And very often we will only be privy to values of $f(x)$ — say corresponding to experimental measurements, or a complicated mathematical expression, or perhaps even the solution of an ordinary differential equation — at a finite number of points x. (Note that here, in our discussion of numerical integration, we do not necessarily require that the values of x at which we know the values of $f(x)$ are the endpoints of the segments, though this is certainly one case of interest.) To address these two issues, we shall

respectively assume that $f(x)$ is roughly constant over each segment $S_i, 1 \le i \le N-1$, and that furthermore (only) the values $f(x_i), 1 \le i \le N-1$, are available.

This then leads naturally to the following approximation. To begin, we consider the integral over S_1: we write

$$\int_{S_1} f(x)\,dx \approx \int_{x_1}^{x_2} f(x_1)dx = (x_2 - x_1)f(x_1) = hf(x_1)\ ;$$

recall that h is the length of each segment. Extending this simple rule to all the segments, we obtain, for $1 \le i \le N-1$,

$$\int_{S_i} f(x) \approx \int_{x_i}^{x_{i+1}} f(x_i)\,dx = (x_{i+1} - x_i)f(x_i) = hf(x_i)\ .$$

We then sum the integrals over all the segments to obtain our approximation to I, which we denote I_h:

$$I_h \equiv hf(x_1) + hf(x_2) + \cdots + hf(x_{N-1})$$

or

$$I_h = \sum_{i=1}^{N-1} hf(x_i)\,. \tag{1}$$

The procedure we have just described is an example of a *quadrature rule* or *numerical integration scheme*. In particular, (1) is called the *rectangle, left rule*, as the approximation is based on the rectangles each of whose height is specified by the function value at the left endpoint of the associated segment. We shall refer to (1) as the "single-sum" version of our quadrature rule.

CYAWTP 2. Include in the sketch you sketched for **CYAWTP 1** the area which corresponds to the term $\int_{S_2} f(x_2)\,dx$.

We now apply our simple quadrature rule (1) to our ocean example to estimate the pressure at the depth of interest $d = 200m$,

$$I(200\text{m}) = \int_{0\text{m}}^{200\text{m}} \gamma(x)dx\ , \tag{2}$$

based on the (depth, specific weight) data of Table 1. We identify $a \equiv 0$m, $b \equiv 200$m, and choose (given the data available in Table 1) $x_1 = 0$m, $x_2 = 100$m, and $x_3 = 200$m associated to intervals $S_1 = (0, 100)$m and $S_2 = (100, 200)$m each of length $h = 100$m. We thus obtain $I_h(200\text{m}) = hf(x_1) + hf(x_2) = 100 \cdot 10.055 + 100 \cdot 10.059 = 2011.4$ kPa.

CYAWTP 3. Apply the simple quadrature rule (1) to our ocean example to estimate the pressure at a depth of 400m based on the (depth, specific weight) data of Table 1.

We emphasize the important role of *discretization*: we may replace $f(x)$ by a constant over each segment — to evaluate the necessary integrals — only because we have first broken the full interval (a, b) into many small segments — to ensure sufficient accuracy. The discretization process

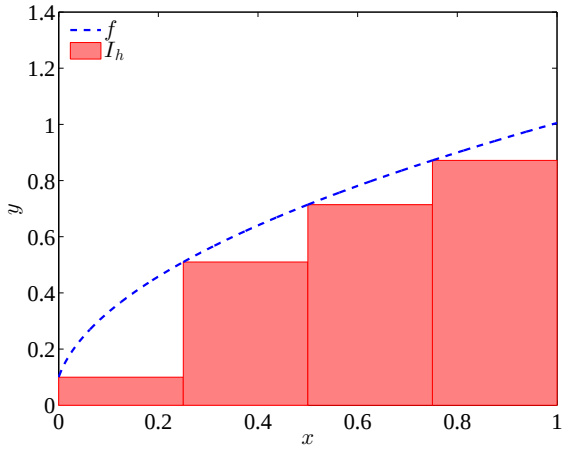

Figure 3: Illustration of the rectangle, left rule over the interval $(a \equiv 0, b \equiv 1)$.

is depicted in Figure 3. We expect that, as we increase the number of segments, our "constant-over-each-segment" approximation $f(x) \approx f(x_i)$ for $x \in S_i$, and hence our "sum-of areas-of-rectangles" approximation $I_h \approx I$, shall be better and better. But more segments and smaller h, and thus higher accuracy, will of course increase computational cost: we observe from (1) that the number of FLoating-point OPerations (FLOPs) required to calculate I_h will scale linearly with $N - 1$; the operation count is $\mathcal{O}(N)$.

It is thus clear that to understand the performance of a scheme — the number of operations required to achieve some desired accuracy — we must investigate the convergence rate, or *order*, of the integration scheme: how quickly does $|I - I_h| \to 0$ as $h \to 0$ (and $N \to \infty$)? Very roughly, a high order scheme will achieve a desired accuracy for larger segments and hence few function evaluations — and hence lower cost. We may then further ask a more practical question: how do we construct integration schemes to provide higher order convergence rates? To answer these questions, we develop a more general framework.

4 A General Framework

We shall consider a particular framework, "integration through interpolation," which builds directly on the construction and analysis of interpolation schemes.

To begin, we return to the "decomposition" of our integral

$$I = \int_a^b f(x)\,dx = \int_{S_1} f(x)\,dx + \int_{S_2} f(x)\,dx + \ldots + \int_{S_{N-1}} f(x)\,dx = \sum_{i=1}^{N-1} \int_{S_i} f(x)\,dx\,.$$

As before, our concerns are twofold: we may not be able to perform the integrals (over each segment) in closed form, even though we are able to evaluate $f(x)$; we may not have access to $f(x)$ except at some finite number of points on the interval (a, b). Thus, over each segment, we shall replace f with an interpolant of f over the segment. Why? First, because the interpolant involves only low-order polynomials, and hence we can easily perform the resulting integrations. Second, because the interpolant only requires function values at some relatively few points. And third, because the

interpolant will be increasingly accurate as we take more and more (smaller and smaller) segments — thus providing convergence.

We denote our interpolant over segment S_i as $\mathcal{I}_i f$. Our approximation procedure can then be summarized as

$$I = \int_a^b f(x)\,dx = \sum_{i=1}^{N-1} \int_{S_i} f(x)\,dx \approx \sum_{i=1}^{N-1} \int_{S_i} (\mathcal{I}_i f)(x)\,dx$$

we thus define our approximation to I, denoted I_h, as

$$I_h \equiv \sum_{i=1}^{N-1} \int_{S_i} (\mathcal{I}_i f)(x)\,dx \ .$$

We recall the two ingredients to an interpolation scheme: the "what" — the form of the interpolant, in our case the order of the polynomial; the "where" — the (interpolation) points at which $(\mathcal{I}f)(x) = f(x)$. Different choices for the interpolant will yield different quadrature rules, as we shall illustrate below. Note we can view $\mathcal{I}_i f$ as a global interpolant over $[a, b]$, but restricted to (evaluated only for) points x in S_i, *or* we can view $\mathcal{I}_i f$ as a local interpolant defined by points on S_i — quadrature points, as we introduce below.

Note that although we take advantage of our segments to motivate and derive our quadrature rule, we may also express (for our family of interpolants) the final result in a single-sum form,

$$I_h = \sum_{i=1}^{N_{\text{eval}}} \omega_i f(\tilde{x}_i) \tag{3}$$

where N_{eval} is the number of quadrature points and weights, the ω_i, $1 \le i \le N_{\text{eval}}$, are the quadrature weights, and the $\tilde{x}_i, 1 \le i \le N_{\text{eval}}$, are the quadrature points — the points at which the function f is evaluated. Note for our simple example of the previous section, the final result, (1), is provided in single-sum form, (3), for suitable choice of quadrature weights and points.

We now wish to understand why this "integration through interpolation" approach provides convergent approximations $I_h \to I$ as $h \to 0$. We recall that, for now, all segments are of the same

length, $h = (b-a)/N$, or equivalently $x_{i+1} - x_i = h, 1 \le i \le N-1$. We may then write

$$\begin{aligned}
|I - I_h| &= |\sum_{i=1}^{N-1} \int_{S_i} f(x)\,dx - \sum_{i=1}^{N-1} (\mathcal{I}_i f)(x)\,dx\,| \\
&\le |\sum_{i=1}^{N-1} \int_{S_i} (f(x) - (\mathcal{I}_i f)(x)\,)\,dx\,| \\
&\le \sum_{i=1}^{N-1} |\int_{S_i} (f(x) - (\mathcal{I}_i f)(x)\,)dx\,| \\
&\le \sum_{i=1}^{N-1} \int_{S_i} |\,f(x) - (\mathcal{I}_i f)(x)\,|\,dx \\
&\le \sum_{i=1}^{N-1} \max_{x\in S_i} |\,f(x) - (\mathcal{I}_i f)(x)\,| \underbrace{\int_{S_i} dx}_{h\equiv(b-a)/(N-1)} \\
&\le e_{\max} \sum_{i=1}^{N-1} h = (N-1)\,h\,e_{\max} = (b-a)\,e_{\max}\ ,
\end{aligned}$$

where $e_{\max}$ is the error in the interpolant over the interval (a,b),

$$e_{\max} \equiv \max_{i=\{1,\ldots,N-1\}} \max_{x\in S_i} |\,f(x) - (\mathcal{I}_i f)(x)\,|\ .$$

In the Interpolation nutshell, for different interpolation choices, we provide bounds for $e_{\max}$ of the form $C_f h^{p_{\mathcal{I}}}$: C_f will depend on the derivatives of the function we wish to interpolate, and in our current discussion, integrate; $p_{\mathcal{I}}$ is the *order*, or convergence rate, of the interpolation scheme (denoted simply p in the Interpolation nutshell). We thus arrive finally at

$$|I - I_h| \le (b-a)\,C_f\,h^{p_{\mathcal{I}}}\ , \tag{4}$$

which is a bound for the error in our approximation I_h.

We may conclude that $I \to I_h$ as $h \to 0$; furthermore, the convergence is at least of order $p_{\mathcal{I}}$. Why do we say "at least"? There are quite a few absolute value signs in our derivation of the bound for $|I - I_h|$, (4), and we might suspect that our bound will not be very precise. We recall that we say that an error bound is sharp if there exists a function for which the bound is exact — the actual error equals the error bound. In general, (4) will not be sharp. In many cases it is just the constant in (4) (for a particular function f), $(b-a)C_f$, which is pessimistic. However, in some other (more special) cases, even the order $p_{\mathcal{I}}$ in (4) is pessimistic: the rate $p_{\int}$ at which the error in the integral converges will be *larger* than the rate $p_{\mathcal{I}}$ at which the interpolant converges; we shall encounter such an instance below.

5 Common Schemes

We present in this section a few common schemes based on the general formulation described above. In many cases it suffices to specify the interpolant and then just "turn the crank" to derive and

subsequently analyze the quadrature formula.

Rectangle, left rule. We re-derive the rectangle, left rule but now as a special case of the general framework; recall we continue to assume here that the segments are all of the same length, h. Here we choose for our interpolant the piecewise-constant, left-endpoint interpolant: hence over each segment S_i, (i) our interpolant is a constant function, and (ii) our single interpolation point in the segment S_i is the left endpoint of the segment, x_i. (In fact, S_i is an open interval, so the interpolation points are chosen in the sense of the limit from within the segment.) The geometric picture — and the motivation for the name "rectangle rule" — is developed in **CYAWTP 2**. The single-sum form, (3), for the resulting quadrature formula is given by (1): $N_{\text{eval}} = N - 1$; $\omega_i = h, 1 \leq i \leq N-1$; $\tilde{x}_i = x_i, 1 \leq i \leq N-1$.

We know from the Interpolation nutshell that for piecewise-constant, left-endpoint interpolation,

$$e_{\max} \leq \max_{i \in \{1,\ldots,N-1\}} \max_{x \in S_i} |f'(x)| \, h \; ;$$

and thus from (4)

$$|I - I_h| \leq (b-a) \max_{i \in \{1,\ldots,N-1\}} \max_{x \in S_i} |f'(x)| \, h \; .$$

Recall that $f'(x)$ denotes the first derivative of f evaluated at x. In fact, this bound for $|I - I_h|$ provides the correct order of convergence, however the constant is not sharp. With some small additional effort we can derive

$$|I - I_h| \leq \frac{b-a}{2} \max_{i \in \{1,\ldots,N-1\}} \max_{x \in S_i} |f'(x)| \, h \; , \tag{5}$$

which now *is* sharp: there exists a function for which the actual error equals our error bound.

CYAWTP 4. Provide a function f for which the rectangle, left rule error bound, (5), holds with equality.

Numerical Experiment 5. Invoke the integration GUI for the function you have proposed in **CYAWTP 4** and confirm your claim.

Important to note that the rectangle, left rule is a *first*-order scheme: it follows from (5) that $p_{\int} = 1$. This implies that our *logarithmic convergence curve* — $\log_{10}(|I - I_h|)$ as a function of $\log_{10}(1/h)$ — asymptotes to a *logarithmic convergence asymptote* with a slope of $-p_{\int} = -1$ for h sufficiently small. In general, we will not observe the asymptotic convergence rate until we *resolve* all the features of f — what happens in f between the quadrature points will not be seen in our approximation I_h.

CYAWTP 6. Consider the functions $\sin(\pi x)$ and $\sin(10\pi x)$. On the same plot, sketch the logarithmic convergence curve and logarithmic convergence asymptote for the rectangle, left rule applied to these two functions.

Numerical Experiment 7. Invoke the integration GUI to confirm your sketch of **CYAWTP 6**.

Rectangle, right rule. Here we choose for our interpolant the piecewise constant, right endpoint interpolant: hence over each segment S_i, (i) our interpolant is a constant function, and (ii) our single interpolation point in segment S_i is the right endpoint of the segment, x_{i+1}. The scheme is quite similar to rectangle, left rule in terms of geometric interpretation and first-order convergence rate.

CYAWTP 8. We are given the monotonically increasing function $f(x) = \exp(x) - 1$ over the interval (a, b). We would like to provide an approximation I_h which converges to $I \equiv \int_a^b f(x)\,dx$ as h tends to zero but also satisfies $I_h > I$ for any h — in other words, I_h is an upper bound for I. Should you apply rectangle, left rule or rectangle, right rule?

Numerical Experiment 9. Invoke the integration GUI to confirm your choice in **CYAWTP 8**.

Rectangle, middle rule. (Note this rule is also referred to more commonly as the *midpoint scheme*.) We choose for our interpolant the piecewise-constant, midpoint interpolant: hence over each segment S_i, (i) our interpolant is a constant function, and (ii) our single interpolation point in segment S_i is the midpoint of the segment, $\tilde{x}_i = (1/2)(x_i + x_{i+1})$. Note that we can understand this choice of interpolant from the local interpolation perspective of the Interpolation nutshell: we choose $\bar{x}_1 = x_i$, $\bar{x}_2 = 1/2(x_i + x_{i+1})$, and $\bar{x}_3 = x_{i+1}$, which defines our interval $S_i = (x_i, x_{i+1})$; we then choose our interpolant as "what": constant, and "where": the point $\bar{x}_2$. (Note that, with these same three points $\bar{x}_1$, $\bar{x}_2$, and $\bar{x}_3$, we could also choose the interpolant "what": quadratic polynomial, and "where": $\bar{x}_1$, $\bar{x}_2$, and $\bar{x}_3$ — we need all three points to uniquely determine our quadratic. This choice for the interpolant yields a well-known numerical integration scheme: *Simpson's rule*. Of course, the Simpson's rule requires function values at all three points, $\bar{x}_1$, $\bar{x}_2$, and $\bar{x}_3$; the rectangle, middle rule requires function value at only the midpoint, $\bar{x}_2$.)

It remains only to derive the quadrature weights. Towards that end, we note that, for our choice of interpolant, $\int_{S_i} (\mathcal{I}_i f)(x)\,dx = \int_{S_i} f(\tilde{x}_i)\,dx = h f(\tilde{x}_i)$. The single-sum form (3) of rectangle, middle rule thus corresponds to the choices $N_{\text{eval}} \equiv N - 1$, $\omega_i = h, 1 \le i \le N_{\text{eval}}$, and $\tilde{x}_i = (1/2)(x_i + x_{i+1}), 1 \le i \le N_{\text{eval}}$. We see that the rectangle, middle rule appears very similar to the rectangle, left rule, with just a slight shift in the quadrature points. This shift, however, substantially improves the accuracy.

CYAWTP 10. Adapt your sketch of **CYAWTP 2** to now reflect the geometric picture — interpretation in terms of areas — associated with rectangle, middle rule. Based on your sketch, do you anticipate that rectangle, middle rule will be more accurate, or less accurate, than rectangle, left rule?

Numerical Experiment 11. Invoke the integration GUI (say) for the function $f(x) = \exp(x) - 1$ to confirm your claim in **CYAWTP 10**.

We now proceed with the error analysis of rectangle, middle rule. If we apply the general error estimate (4), we would predict a first-order scheme. However, a more precise analysis — which reflects your intuition of **CYAWTP 10** — yields the sharp estimate

$$|I - I_h| \le \frac{b-a}{24} \max_{i \in \{1,\ldots,N-1\}} \max_{x \in S_i} |f''(x)|\, h^2 \ .$$

Recall that $f''(x)$ denotes the second derivative of f evaluated at x. Important to note that the rectangle, middle rule is a *second*-order scheme, $p_{\int} = 2$: if we double the number of function

evaluations, the error decreases by a factor of four (as $h \to 0$). Note in particular that the order of integration, $p_{\int} = 2$, is larger than the order of interpolation, $p_{\mathcal{I}} = 1$. (The midpoint scheme is an example of a *Gauss quadrature* rule in which the integration weights *and* points are chosen to optimize the convergence rate.)

Trapezoidal rule. Here we choose for our interpolant the (global) piecewise-linear interpolant associated with the discretization $x_1, x_2, \ldots, x_N$. The name of this scheme is not a coincidence.

CYAWTP 12. Adapt your sketch of **CYAWTP 2** to now reflect the geometric picture — interpretation in terms of areas — associated with trapezoidal rule. Based on your figure, do you anticipate that the trapezoidal rule will be more accurate, or less accurate, than rectangle, left rule?

In order to develop our quadrature rule, we first consider a particular segment, S_i. We recall that over segment S_i our piecewise-linear interpolant reduces to

$$(\mathcal{I}_i f)(x) = f(x_i) + \left(\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \right) (x - x_i) \quad x \in S_i \; ;$$

it thus follows that

$$\int_{S_i} (\mathcal{I}_i f)(x) dx \equiv \int_{x_i}^{x_{i+1}} \left(f(x_i) + \left(\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \right) (x - x_i) \right) dx = h \, \frac{1}{2} \left(f(x_i) + f(x_{i+1}) \right) .$$

To derive this result we can perform the simple integrations explicitly. More simply, we can appeal to your sketch of **CYAWTP 12** and note that the area of a trapezoid of base h and left and right heights $f(x_i)$ and $f(x_{i+1})$, respectively, is given by $h \, (1/2) \, (f(x_i) + f(x_{i+1}))$. We can now readily assemble the contributions from each segment: our recipe gives

$$I_h = \sum_{i=1}^{N-1} \int_{S_i} (\mathcal{I}_i f)(x) \, dx = \sum_{i=1}^{N-1} h \, \frac{1}{2} \left(f(x_i) + f(x_{i+1}) \right). \tag{6}$$

Recall that this final result reflects our assumption of equispaced points.

CYAWTP 13. Derive the single-sum quadrature formula for the trapezoidal rule. In particular, in (3), for the trapezoidal rule, how must we choose N_{eval}? the quadrature weights, $\omega_i, 1 \le i \le N_{\text{eval}}$? the quadrature points, $\tilde{x}_i, 1 \le i \le N_{\text{eval}}$?

We know from the Interpolation nutshell that for piecewise-linear interpolation,

$$e_{\max} \le \frac{1}{8} \max_{i \in \{1,\ldots,N-1\}} \; \max_{x \in S_i} \; |f''(x)| \, h^2 \; .$$

and thus

$$|I - I_h| \le \frac{b-a}{8} \max_{i \in \{1,\ldots,N-1\}} \; \max_{x \in S_i} \; |f''(x)| \, h^2 \; .$$

(This bound can be sharpened slightly: the (1/8) may be replaced with (1/12).) Important to note that the "trapezoidal rule" is a *second*-order scheme: if we double the number of function evaluations, the error decreases by a factor of four (as $h \to 0$). Recall that this bound will not be applicable if the integrand is not sufficiently differentiable, though in general smoothness is less crucial for integration than for interpolation.

CYAWTP 14. Which, if any, of the schemes rectangle, left rule, rectangle, right rule, rectangle, middle rule, and trapezoidal rule will integrate exactly a linear function $f(x) = mx + c$ (for m and c given constants).

CYAWTP 15. Consider the integral $I = \int_0^1 f(x)dx$ for

$$f(x) = \begin{cases} 0, & 0 \le x < 1/3 \\ 1, & 1/3 \le x \le 1 \end{cases} .$$

Sketch the logarithmic convergence curve and the corresponding logarithmic convergence asymptote for trapezoidal rule for a "doubling" uniform refinement strategy. We recall that the latter corresponds to discretization parameters $h = 1$, $h = 1/2, h = (1/2)^2, h = (1/2)^3, \ldots$, such that the jump in the function at $x = 1/3$ will always reside *inside* a segment and not at a segment endpoint. What is the slope of the logarithmic convergence curve?

Numerical Experiment 16. Invoke the integration GUI to confirm your claims in **CYAWTP 15**. Note that the Integration GUI considers a "doubling" uniform refinement strategy $h = 1$, $h = 1/2$, $h = (1/2)^2$, $h = (1/2)^3$, ..., such that the jump in the function at $x = 1/3$ will always reside *inside* a segment and not at a segment endpoint.

CYAWTP 17. Consider the trapezoidal rule for the more general situation in which the segments $S_i, 1 \le i \le N-1$, are of variable length: for $1 \le i \le N-1$, S_i is of length $h_i \equiv x_{i+1} - x_i$. Repeat **CYAWTP 13** but now for this case of variable-length segments.

CYAWTP 18. Propose functions $f(x)$ for which the variable-length trapezoidal rule (or more generally, variable-length versions of any of our schemes) could prove advantageous in terms of reduced computational cost? Note that your functions $f(x)$ should not depend on the discretization grid (or h).

6 Perspectives

We have only here provided a first look at the topic of numerical integration. A more in-depth study may be found in *Math, Numerics, and Programming (for Mechanical Engineers)*, M Yano, JD Penn, G Konidaris, and AT Patera, available on MIT OpenCourseWare, which adopts similar notation to these nutshells and hence can serve as a companion reference. For an even more comprehensive view from both the computational and theoretical perspectives we recommend *Numerical Mathematics*, A Quarteroni, R Sacco, F Saleri, Springer, 2000.

Of the many further topics of interest, perhaps the most important is the treatment of integration in higher dimensions. In this nutshell we consider integration of a univariate function over an integral. In many engineering situations we must evaluate integrals over surfaces or volumes, for example to assess overall performance metrics related to heat transfer rates or flowrates or forces and moments. In two or three dimensions, the methods described in this nutshell extend quite easily: we replace our small segments with (in two dimensions) small triangles or small rectangles. However, there are also cases in which the domain is very high dimensional. In such cases discretization is simply too expensive, and a very different approach is warranted: Monte Carlo methods. We discuss the latter in a later nutshell.

2.086 Numerical Computation for Mechanical Engineers
Fall 2014