Honors Numerical Analysis

Lecture 21

Richardson Extrapolation

If we are computing some φ using a scheme $\varphi_0(h)$,

then the error can be characterized as:

$$(*)\quad \underbrace{\varphi}_{\text{exact}} = \underbrace{\varphi_0(h)}_{\text{estimate}} + \underbrace{c_1 h + c_2 h^2 + c_3 h^3 + \ldots}_{\text{error.}}$$

If we were to halve h to $h/2$, then we have:

$$(**)\quad \varphi = \varphi_0(h/2) + c_1 \frac{h}{2} + c_2 \frac{h^2}{4} + c_3 h^3/8 + \ldots$$

We could take linear combinations of $(*)$ and $(**)$ to eliminate the $\mathcal{O}(h)$ term:

$$2\varphi - \varphi = 2\varphi_0(h/2) + c_1 h + c_2 \frac{h^2}{2} + c_3 h^3/4 + \ldots$$
$$- \left(\varphi_0(h) + c_1 h + c_2 h^2 + c_3 h^3 + \ldots\right)$$

$$\Rightarrow \varphi = \left(2\varphi_0(h/2) - \varphi_0(h)\right) - c_2 h^2/2 + \ldots$$

$$= \underbrace{\left(2\varphi_0(h/2) - \varphi_0(h)\right)}_{\text{Richardson extrapolated value.}} + \underbrace{\mathcal{O}(h^2)}_{\text{error went down by one order.}}$$

Example The Trapezoidal Rule

We saw that $I = \int_a^b f(x)\,dx$ can be approximated by the trapezoidal rule:

$$I \approx I_0(h) = h\sum_{i=0}^{n} f(x_i) - \frac{h}{2}\left(f(a)+f(b)\right), \quad h = \frac{b-a}{n}$$

The error is $O(h^2)$.

This means that

$$I = I_0(h) + \underbrace{O(h^2)}_{c_2h^2} + \underbrace{O(h^4)}_{c_4h^4} + \ldots$$

this can be shown to be the next term in the composite trapezoidal rule.

And therefore:

$$I = I_0(h/2) + O\left(\frac{h^2}{4}\right) + O(h^4/16) + \ldots$$

Taking

$$4I - I = 4I_0(h/2) + O(h^2) + O(h^4/4) + \ldots - \left(I_0(h) + O(h^2) + O(h^4) + \ldots\right)$$

$$\Rightarrow \quad I = \underbrace{\frac{4I_0(h/2) - I_0(h)}{3}}_{\text{Call this } I_1(h)} + O(h^4)$$

the error went from $O(h^2)$ to $O(h^4)$

The same procedure could be performed with $I_1(h)$ and

$$I_1(h/2) = \frac{4I_0(h/4) - I_0(h/2)}{3}$$

to ~~compute~~ eliminate the $O(h^4)$ error and compute $I_2(h)$.

Applicable to many numerical estimates (Finite differences, etc.)

Gaussian Quadrature Rules

General form of quadrature rules: $\int_a^b f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$.

n weights　　n nodes.

This rule can be made exact for certain functions.

Ex: If $\int_a^b \cos x\,dx = w_j \cos x_j$ then x_j can be anything, and $w_j = \frac{1}{\cos x_j} \int_a^b \cos x\,dx$

= a number.

Once the nodes x_j are determined, finding the correct w_j's such that $f_1, \dots, f_n$ are integrated exactly requires only solving a linear system.

Ex: If the quadrature rule $\int_a^b f(x)\,dx = \sum_{j=1}^{n} w_j f(x_j)$

is exact for $f = f_1, f_2, \dots f_n$ then it must be the case that

$$\sum_{j=1}^{n} w_j f_1(x_j) = \int_a^b f_1(x)\,dx = I_1$$

$$\sum_{j=1}^{n} w_j f_2(x_j) = \int_a^b f_2(x)\,dx = I_2$$

$$\vdots$$

$$\sum_{j=1}^{n} w_j f_n(x_j) = I_n$$

If the nodes are specified, then the rule can be made exact for n functions by solving a linear system.

$\Rightarrow$ This can be written as:

$$\begin{pmatrix} f_1(x_1) & f_1(x_2) & \cdots & f_1(x_n) \\ \vdots & & & \\ f_n(x_1) & & \cdots & f_n(x_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{pmatrix}.$$

A quadrature rule of the form $\int_a^b f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$ is called GAUSSIAN if it is exact for $2n$ linearly independent functions.

How is this possible? Nonlinear optimization/root finding

I.e. find weights and nodes simultaneously so that $2n$ integrals are computed exactly.

Special Case Integrating the functions $\underbrace{1, x, x^2, \ldots, x^{2n-1}}_{\text{polynomials of degree less than or equal to } 2n-1}$ exactly.

The key to deriving Gaussian quadrature rules for polynomials is orthogonal polynomials.

(*) Theorem If $x_1, \ldots, x_n$ are the zeros (roots) of P_n, the degree n Legendre polynomial, then the formula:

$$\int_a^b f(x)\,dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

when $w_j = \int_a^b q_j(x)\,dx$, $\quad q_j(x) = \prod_{l \neq j} \frac{x - x_l}{x_j - x_l}$

is exact for polynomials of degree $2n-1$ or less.

$\Rightarrow$ Exact for $2n$ linearly independent functions!

Back to Gaussian Quadratures:

<u>Proof</u> [of Theorem (*)] Let f be a polynomial of degree $\leq 2n-1$. This implies that $f = q P_n + r$ where

$\deg(q) \leq n-1$, $\deg(P_n) = n$, $\deg(r) \leq n-1$ } by polynomial long division

So, if x_j is a root of P_n, then

$$f(x_j) = q(x_j) \underbrace{\cancel{P(x_j)}}_{=0} + r(x_j) = r(x_j).$$

Now integrate:

$$\int_{-1}^{1} f(x)\,dx = \underbrace{\int_{-1}^{1} q(x)\,P_n(x)\,dx}_{=0 \text{ since } \deg(q) \leq n-1} + \int_{-1}^{1} r(x)\,dx$$

$$= \int_{-1}^{1} r(x)\,dx.$$

Now: given the choice of our weights earlier,

$$\int_{-1}^{1} r(x)\,dx = \sum_{j=1}^{n} w_j\, r(x_j) = \sum_{j=1}^{n} w_j\, f(x_j).$$

But w_j was chosen to give an exact integral for any polynomial of degree $\leq n-1$. (From the Newton-Cotes construction)

To summarize On the interval $[-1,1]$, the n-point Gaussian Quadrature integrates $1, x, x^2, \ldots, x^{2n-1}$ exactly, where the nodes x_j are the roots of P_n and the weights can be precomputed from Newton-Cotes.

Another Example: Chebyshev Polynomials

Chebyshev polynomials are orthogonal $(-1,1)$ with weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$.

Let x_j be the roots of T_n. Then the weights are:

$$w_j = \int_{-1}^{1} q_j(x)\,w(x)\,dx, \quad q_j(x) = \prod_{\ell \neq j} \frac{x - x_\ell}{x_j - x_\ell}.$$

Then the quadrature rule:

$\Rightarrow$ $$\int_{-1}^{1} f(x)\frac{1}{\sqrt{1-x^2}}\,dx \approx \sum_{j=1}^{n} w_j\, f(x_j)$$ is exact for f a polynomial of degree $\leq 2n-1$.