CLASS NOTES

Models, Algorithms and Data: Introduction to computing

2019

Petros Koumoutsakos, Jens Honore Walther, Julija Zavadlav

(Last update: May 17, 2019)

IMPORTANT DISCLAIMERS

Much of the material (ideas, definitions, concepts, examples, etc) in these notes is taken for teaching purposes, from several references:

- Numerical Analysis by R. L. Burden and J. D. Faires
- The Nature of Mathematical Modeling by N. Gershenfeld
- A First Course in Numerical methods by U. M. Ascher and C. Greif

These notes are only informally distributed and intended ONLY as study aid for the final exam of ETHZ students that were registered for the course Models, Algorithms and Data (MAD): Introduction to computing 2019. The notes have been checked, however they may still contain errors so use with care.

LECTURE 10 Numerical integration III: Adaptive Quadrature

While Romberg integration can achieve arbitrary accuracy, it is not the most efficient method in terms of function evaluations. One of the reasons Romberg integration requires many evaluations in some cases (e.g. functions with sharp peaks) is that the intervals are equispaced. Many evaluations take place in areas that are not "important" (i.e. in areas where the function value is small or slowly varying). Intuitively we must use fewer points where the function varies slowly and more at places with strong variations.

10.1 Adaptive Integration

We want to increase the accuracy where needed and stop refining in the other regions. Thus we have to identify the regions where we need refinement. Thinking of the methods learned in the last lecture we realize, that we already derived the formal tool to estimate the error for a given quadrature rule when we talked about Richardson extrapolation. We then found

$$\epsilon(h/2) \approx G(h/2) - G(h) \tag{10.1.1}$$

Using this we can thus define an adaptive integration procedure as follows:

Algorithm 1 Adaptive integration.

Steps:

Subdivide the interval of the integration into sub-intervals
for all sub-intervals **do**
  Compute sub-integral, estimate the error with Richardson procedure described earlier.
  if accuracy is worse than desired **then**
    Subdivide the interval
  else
    Leave the interval untouched
  end if
end for

10.2 Gauss Quadrature

In the above spirit, namely that certain points and regions are more important when evaluating the integral we expect to obtain higher accuracy by adapting the weights and abscissas of our quadrature rules. We want to choose them such that they maximize the accuracy of the formulas. As we will see in the following this can be cast into the problem of solving a non-linear set of equations.

10.2.1 Method of undetermined coefficients

As a warm up, let us re-derive the trapezoidal rule using the method of undetermined coefficients. We start by approximating

$$\int_a^b f(x)dx \approx c_1 f(a) + c_2 f(b). \tag{10.2.1}$$

Let the right hand side of (10.2.1) be exact for integrals of a straight line, e.g.

$$\int_a^b f(x)dx = \int_a^b (a_0 + a_1 x)dx = \left[a_0 x + a_1 \frac{x^2}{2}\right]_a^b = a_0(b-a) + a_1\left(\frac{b^2-a^2}{2}\right). \tag{10.2.2}$$

In particular, for $f(x) = a_0 + a_1 x$, we want right hand sides of (10.2.1) and (10.2.2) to be equal,

$$a_0(b-a) + a_1\left(\frac{b^2-a^2}{2}\right) = c_1(a_0 + a_1 a) + c_2(a_0 + a_1 b), \tag{10.2.3}$$

which can be rewritten as

$$a_0(b-a) + a_1\left(\frac{b^2-a^2}{2}\right) = a_0(c_1 + c_2) + a_1(c_1 a + c_2 b). \tag{10.2.4}$$

Since the above equation needs to be satisfied for arbitrary values of constants a_0 and a_1 for a general straight line, we require

$$c_1 + c_2 = b - a, \qquad c_1 a + c_2 b = \frac{b^2-a^2}{2}. \tag{10.2.5}$$

The above quadratic system of equations can be easily solved, obtaining

$$c_1 = c_2 = \frac{b-a}{2}, \tag{10.2.6}$$

which recovers the trapezoidal rule.

Derivation of two-point Gauss rule

The two-point Gauss quadrature is an extension of the trapezoidal rule approximation, where the arguments of the function are not predetermined as a and b, but considered as unknowns x_1 and x_2. Henceforth, in the two-point Gauss quadrature rule, the integral is approximated as

$$I = \int_a^b f(x)dx \approx c_1 f(x_1) + c_2 f(x_2). \tag{10.2.7}$$

There are four unknowns x_1, x_2, c_1, and c_2. These are found by requiring that the rule (10.2.7) gives exact results for integration of a general third order polynomial, $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$[1]. Hence

$$\begin{aligned}\int_a^b f(x)dx = \int_a^b (a_0 + a_1 x + a_2 x^2 + a_3 x^3)dx &= \left[a_0 x + a_1 \frac{x^2}{2} + a_2 \frac{x^3}{3} + a_3 \frac{x^4}{4}\right]_a^b \\ &= a_0(b-a) + a_1\left(\frac{b^2-a^2}{2}\right) + a_2\left(\frac{b^3-a^3}{3}\right) + a_3\left(\frac{b^4-a^4}{4}\right).\end{aligned} \tag{10.2.8}$$

Alternatively, applying quadrature rule (10.2.7), we obtain

$$\int_a^b f(x)dx \approx c_1 f(x_1) + c_2 f(x_2) = c_1(a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3) + c_2(a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3). \tag{10.2.9}$$

Equating right hand sides of (10.2.8) and (10.2.9), we have

$$\begin{aligned}&a_0(b-a) + a_1\left(\frac{b^2-a^2}{2}\right) + a_2\left(\frac{b^3-a^3}{3}\right) + a_3\left(\frac{b^4-a^4}{4}\right) \\ &= c_1(a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3) + c_2(a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3) \\ &= a_0(c_1 + c_2) + a_1(c_1 x_1 + c_2 x_2) + a_2(c_1 x_1^2 + c_2 x_2^2) + a_3(c_1 x_1^3 + c_2 x_2^3).\end{aligned} \tag{10.2.10}$$

Since in (10.2.10), the constants a_0, a_1, a_2, and a_3 are arbitrary, the respective coefficients must be equal as well, resulting in the following cubic system equations for c_1, c_2, x_1, and x_2:

$$\begin{aligned}b - a &= c_1 + c_2, \\ \frac{b^2-a^2}{2} &= c_1 x_1 + c_2 x_2, \\ \frac{b^3-a^3}{3} &= c_1 x_1^2 + c_2 x_2^2, \\ \frac{b^4-a^4}{4} &= c_1 x_1^3 + c_2 x_2^3.\end{aligned} \tag{10.2.11}$$

Without proof, we can find the above four simultaneous nonlinear equations have only one acceptable solution

$$\begin{aligned}c_1 &= \frac{b-a}{2}, \\ c_2 &= \frac{b-a}{2}, \\ x_1 &= \left(\frac{b-a}{2}\right)\left(-\frac{1}{\sqrt{3}}\right) + \frac{b+a}{2}, \\ x_2 &= \left(\frac{b-a}{2}\right)\left(\frac{1}{\sqrt{3}}\right) + \frac{b+a}{2}.\end{aligned} \tag{10.2.12}$$

[1] As we now consider the abscissa to be unknowns we obtain two additional degrees of freedom and thus we can consider a cubic polynomial instead of the linear one from before

Solution values for c_1, c_2, x_1, and x_2 as in (10.2.12), inserted into (10.2.7), lead to the **two-point Gauss quadrature rule**:

$$\int_a^b f(x)dx \approx \frac{b-a}{2} f\left[\left(\frac{b-a}{2}\right)\left(-\frac{1}{\sqrt{3}}\right)+\frac{b+a}{2}\right]+\frac{b-a}{2} f\left[\left(\frac{b-a}{2}\right)\left(\frac{1}{\sqrt{3}}\right)+\frac{b+a}{2}\right]. \tag{10.2.13}$$

Higher order Gauss quadrature rules and their properties are better understood in terms of Hermite interpolation and Legendre polynomials and their properties.

10.2.2 Hermite Interpolation

In order to derive n-point Gauss quadrature rules we have to introduce a new case of interpolation: Hermite interpolation. This is an interpolation that allows for extra degrees of smoothness than Lagrange interpolation. Recall for Lagrange interpolation we have that:

- Lagrange interpolants tend to oscillate about the exact function.
- Smooth functions are interpolated more accurately than the ones that oscillate or have concentrated curvature.
- Decrease in error as the number of points and degree of polynomials increase depends strangely on the function nature.
- Extrapolation yields much larger errors than interpolation.

Returning to the requirement of enhanced smoothness, one way to interpolate a function is not only to interpolate its function values but also to interpolate its derivatives.

Given y_i, y_i' at $i = 1,\dots,n$ data points $x_1,\dots,x_n$ find a polynomial $f(x)$ of degree $2n-1$ that fits all the data (i.e. $y_i = f(x_i)$ and $y_i' = f'(x_i)$):

$$\boxed{f(x) = \sum_{k=1}^{n} U_k(x) y_k + \sum_{k=1}^{n} V_k(x) y_k',} \tag{10.2.14}$$

where U_k and V_k are polynomials of degree $2n-1$ with the following properties:

$$U_k(x_j) = \delta_{jk}, \qquad U_k'(x_j) = 0, \tag{10.2.15}$$

$$V_k(x_j) = 0, \qquad V_k'(x_j) = \delta_{jk}. \tag{10.2.16}$$

The required polynomials can be constructed using properties of the Lagrange polynomials $L_k(x)$:

$$L_k(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_1)(x_k-x_2)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} \tag{10.2.17}$$

The resulting U_k and V_k can then be expressed in terms of $L_k(x)$ as follows:

$$U_k(x) = \left[1 - 2L'_k(x_k)(x - x_k)\right] L_k^2(x), \tag{10.2.18}$$

$$V_k(x) = (x - x_k)L_k^2(x). \tag{10.2.19}$$

By the properties of the Lagrange polynomials we can check that the so constructed functions satisfy the wished properties. The polynomials are smoother and more accurate but suffer from some of the pathologies of Lagrange polynomials.

10.2.3 n-point Gauss rule

As is conventional, we will move from our general integration interval $I = (a, b)$ to $I' = (-1, 1)$. In order make use of the found weights and abscissas we have to perform a transformation of variables

$$\boxed{\xi = \frac{2x - (a + b)}{b - a}} \tag{10.2.20}$$

So any x in (a, b) is transformed to an integral in ξ on $(-1, 1)$.

Let us now use the found interpolation to obtain $\int_{-1}^{1} f(x)\mathrm{d}x$. As done previously we approximate f by Hermite polynomials (Eq. (10.2.14))

$$\int_{-1}^{1} f(x)dx = \sum_{k=1}^{n} y_k \int_{-1}^{1} U_k(x)dx + \sum_{k=1}^{n} y'_k \int_{-1}^{1} V_k(x)dx, \tag{10.2.21}$$

Which can be rewritten as follows:

$$\int_{-1}^{1} f(x)dx = \sum_{k=1}^{n} u_k f(x_k) + \sum_{k=1}^{n} v_k f'(x_k), \tag{10.2.22}$$

where:

$$u_k = \int_{-1}^{1} U_k(x)dx, \qquad v_k = \int_{-1}^{1} V_k(x)dx. \tag{10.2.23}$$

In order for equation 10.2.22 to be of the form $I = \int_{-1}^{1} f(x)\mathrm{d}x = \sum_{i=1}^{n} w_i f(x_i)$ we must make $v_k = 0$ for all k. This can be assured by choosing accordingly the abscissas. Inserting the expression for $V_k(x)$ we find

$$v_k = \int_{-1}^{1} (x - x_k)L_k^2(x)dx. \tag{10.2.24}$$

Now recall that we can decompose the Lagrange polynomials as $L_k(x) = C_k F(x)/(x - x_k)$ with

$$C_k = \frac{1}{(x_k - x_1)(x_k - x_2)\cdot\ldots\cdot(x_k - x_{k-1})(x_k - x_{k+1})\cdot\ldots\cdot(x_k - x_n)}, \tag{10.2.25}$$

$$F(x) = (x - x_1)(x - x_2)\cdot\ldots\cdot(x - x_{n-1})(x - x_n). \tag{10.2.26}$$

Inserting this factorisation for one of the L_k's we find

$$v_k = C_k \int_{-1}^{1} F(x) L_k(x)\, dx. \tag{10.2.27}$$

As C_k is non-zero we have to make sure that the following integral if it is zero for all k

$$\boxed{0 = \int_{-1}^{1} F(x) L_k(x)\, dx} \tag{10.2.28}$$

We know that $F(x)$ is a polynomial of degree n, where $L_k(x)$ is a polynomial of degree $n-1$. If the abscissas x_k are all different, the Lagrange polynomials form a linearly independent set. Hence Eq. (10.2.28) says that $F(x)$ is a polynomial of degree n that is orthogonal to any polynomial of degree $n-1$. This polynomial is unique and is well known to be the Legendre polynomial of degree n, $P_n(x)$. The Legendre polynomials have this orthogonality property:

$$\int_{-1}^{1} P_n(x) P_m(x)\, dx = N_n \delta_{nm}, \tag{10.2.29}$$

where $N_n = 2/(2n+1)$ is a normalization constant. So this analysis shows that the abscissas we try to find are the zeros of the Legendre polynomial of degree n; it is possible to show that the zeros lie between -1 and $+1$.

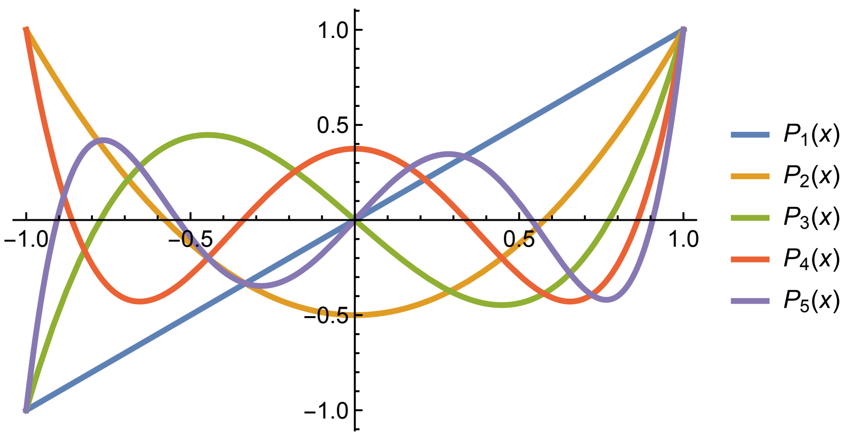

Figure 10.1: Legendre polynomials $P_n(x)$ for $n = 1, \ldots, 5$.

The weights can be computed from Eq. (10.2.23) for the given abscissa so we have that

$$\boxed{\int_{-1}^{1} f(x)\, dx = \sum_{k=1}^{n} u_k f(x_k)} \tag{10.2.30}$$

We compute x_k as the k-th root of $P_n(x)$ and one can show that the weights in Eq. (10.2.30) are given by (according to *Handbook of Mathematical Functions* by Abramowitz and Stegun):

$$u_k = \frac{2}{\left(1 - x_k^2\right)(P_n'(x_k))^2}. \tag{10.2.31}$$

The values of x_k and the respective weights for $n = 8$ can be found in the following table

Points x_k	-0.96029	-0.796666	-0.525532	-0.183435	0.183435	0.525532	0.796666	0.96029
Weights u_k	0.101229	0.222381	0.313707	0.362684	0.362684	0.313707	0.222381	0.101229

Error The error with n abscissas is:

$$\varepsilon = \frac{2^{2n+1}(n!)^4}{(2n+1)(2n!)^3} f^{(2n)}(\xi). \tag{10.2.32}$$

Gauss Quadrature gives the best accuracy (in the sense of correctly integrating polynomials of highest possible order for a given number of function evaluations). Abscissas for various orders are all different. If one wishes to improve the accuracy, new calculations must be done from scratch. Hence for some cases Romberg or adaptive integration are still "better".

Exam checklist

After this class, you should understand the following points regarding numerical integration:

- How to improve accuracy of integration locally with adaptive quadrature
- How to choose the locations of function evaluation optimally with Gauss quadrature