Integration with Gaussian quadratures

Xavier R. Hoffmann

August 29, 2021

---

1 Introduction

2 Theoretical elements

Our goal is to numerically calculate the definite integral

$$I = \int_a^b f(x)\,dx \tag{1}$$

with f a well-behaved function (smooth and non-diverging) in the interval $[a, b]$. The **mean value theorem for integrals**[1] allows us to write the integral I as a sum of N terms

$$I = \sum_{i=1}^{N} \tilde{w}_i f(x_i) \qquad x_i \in [a, b]\ \forall i \tag{2}$$

where $\{x\}$ are the **integration nodes** and $\{\tilde{w}\}$ the **weights**. Solving the above equality would mean finding the exact values of $\{x\}$ and $\{\tilde{w}\}$, for *all* f. In theory, this could be possible, but then we'd have an analytical solution to the problem, thus no need for a numerical method.

Developing a numerical integration method consists in requiring that the equality holds exactly for *some* f and finding suitable values for the nodes and weights. For other functions, the sum will then be an *approximation* to the integral, so it's formally more correct to write: $I \simeq \sum_{i=1}^{N} \tilde{w}_i f(x_i)$. Note that we'll need $2N$ conditions to solve the $2N$ unknowns (N nodes and N weights).

As an example, let's analyze the popular trapezoid rule, in which we impose that $x_1 = a$, $x_N = b$ and that all nodes be equidistant. These N conditions for the nodes yield the familiar[2] $x_i = a + (i-1)\cdot h$, with $h = (b-a)/(N-1)$. On the other hand, we require the method to be *exact* for polynomials of degree zero *and* for polynomials of degree one, which translates to N additional conditions. These allow us to find the values for the weights: $\tilde{w}_1 = \tilde{w}_N = h/2$ and $\tilde{w}_i = h, i \in \{2, 3, ..., N-1\}$.

---

[1] see appendix A

[2] Beware of the notation! N is the total number of nodes, not subintervals.

2.1 Gaussian quadratures

Before continuing we introduce a small change of notation[3], $f = g \cdot v$. We're also slightly changing the equality between integral and sum[4]

$$\int_a^b g(x)v(x)\,dx \simeq \sum_{i=1}^{N} w_i g(x_i)\ . \tag{3}$$

Remember that our problem consists in solving $2N$ unknowns (N nodes and N weights), so we'll need $2N$ conditions. In Gaussian quadratures, we demand our solution to be *exact* for any polynomial T, with $0 \leq \deg(T) \leq 2N-1$. *Habemus* our $2N$ conditions!

Let $\{\phi\}$ be an **orthogonal polynomial basis** of $L^2_v[a,b]$

$$\langle \phi_k | \phi_l \rangle = \int_a^b \phi_k(x)\phi_l(x)v(x)\,dx = A_k^2 \delta_{k,l} \tag{4}$$

with A_k^2 the normalization constant and $v(x)$ the weight function[5]. Additionally we demand that ϕ_k has k **simple roots** in $[a,b]$.

Now we take *any* polynomial of degree $N-1$, R_{N-1}, which can be expanded (in the interval $[a,b]$) using the chosen orthogonal basis

$$R_{N-1}(x) = \sum_{k=0}^{N-1} \alpha_k \phi_k(x)\ , \tag{5}$$

and construct the function $g = \phi_N R_{N-1}$. Note that g is a polynomial of order $2N-1$; if we calculate its integral, the equality in (3) will hold *exactly*

$$\int_a^b \phi_N(x)R_{N-1}(x)v(x)\,dx = \sum_{i=1}^{N} w_i \phi_N(x_i) R_{N-1}(x_i)\ . \tag{6}$$

Writing out the LHS, using (4) and (5)

$$\int_a^b \phi_N(x)R_{N-1}(x)v(x)\,dx = \sum_{k=0}^{N-1} \alpha_k \int_a^b \phi_N(x)\phi_k(x)v(x)\,dx = \sum_{k=0}^{N-1} \alpha_k A_k^2 \delta_{k,N} = 0\ . \tag{7}$$

The RHS of (6) must also sum zero. In general, $R_{N-1}(x_i) \neq 0$ and, obviously, $w_i \neq 0$; thus: $\phi_N(x_i) = 0$.

First result: Our integration nodes will be the roots of the N-th degree polynomial of the $L^2_v[a,b]$ orthogonal basis $\{\phi\}$.

[3] If f is well-behaved, there shouldn't be any problem in obtaining this decomposition.

[4] You might be alarmed by the fact that we're messing with the mean value theorem, but we've only changed the weights (there's a map from the previously defined weights to these new ones).

[5] Not to be confused with our weights $\{w\}$!

2.2 Lagrange interpolation

So far we've solved the N nodes $\{x_i\}$, using an $L^2_v[a,b]$ orthogonal polynomial basis $\{\phi\}$. We still have to find the value for the N weights $\{w_i\}$; to do so we use **Lagrange interpolation**.

Consider *any* function h and its Lagrange interpolator H_{N-1}, a polynomial of degree $N-1$. In the interval $[a,b]$, we want them to coincide at N points, specifically the N roots of ϕ_N, $\{x_j\}$. We define H_{N-1} as

$$H_{N-1}(x) = \sum_{j=1}^{N} h(x_j)\frac{\phi_N(x)}{\phi'_N(x_j)(x-x_j)} \tag{8}$$

with ϕ'_N the derivative of ϕ_N. One might be worried about H_{N-1} diverging (we're apparently dividing by zero!). Remember that any polynomial can be written as a product of roots, particularly

$$\phi_N = c_N(x-x_1)(x-x_2)\cdots(x-x_N) \tag{9}$$

which makes it easy to check that for each term of the sum, the apparent zero in the denominator cancels out with a term in the numerator[6]. On the other hand, remember that ϕ_N has N simple roots, which implies[7] $\phi'_N(x_j) \neq 0, \forall j$. Let's check if the two functions h and H_{N-1} coincide at the N roots[8]

$$H_{N-1}(x_k) = \lim_{x\to x_k} H_{N-1}(x) = \sum_{j=1}^{N} h(x_j) \lim_{x\to x_k} \frac{\phi_N(x)}{\phi'_N(x_j)(x-x_j)} = \sum_{j=1}^{N} h(x_j)\delta_{j,k} = h(x_k)\,. \tag{10}$$

Now we particularize this expression for a polynomial of degree $2N-1$, q_{2N-1}, whose Lagrange interpolator we denote Q_{N-1}. Remember that q_{2N-1} and Q_{N-1} coincide at N points, the roots of ϕ_N. We can write the difference $s = q_{2N-1} - Q_{N-1}$ (note that $\deg(S) = 2N-1$) as

$$\begin{aligned} s(x) &= q_{2N-1}(x) - Q_{N-1}(x) && (11)\\ &= B(x-x_1)(x-x_2)\cdots(x-x_N)(x-\tilde{x}_1)(x-\tilde{x}_2)\cdots(x-\tilde{x}_{N-1})\,. && (12) \end{aligned}$$

The first N roots of s coincide with those of ϕ_N, the remaining $N-1$ roots isn't relevant. We define one last polynomial

$$r(x) = \frac{s(x)}{\phi_N(x)} = \frac{q_{2N-1}(x) - Q_{N-1}(x)}{\phi_N(x)} = B(x-\tilde{x}_1)(x-\tilde{x}_2)...(x-\tilde{x}_{N-1})\,, \tag{13}$$

with $\deg(r) = N-1$. Rearranging expression (13)

$$q_{2N-1} = Q_{N-1} + r\phi_N\,. \tag{14}$$

Finally we identify q_{2N-1} with the function g and, as we've done before, calculate its integral. Again, the equality in (3) will hold *exactly*

$$\int_a^b q_{2N-1}(x)v(x)\,dx = \sum_{i=1}^{N} w_i q_{2N-1}(x_i)\,. \tag{15}$$

Using (14) we write out the LHS in (15)

$$\int_a^b q_{2N-1}(x)v(x)\,dx = \int_a^b Q_{N-1}(x)v(x)\,dx + \int_a^b r(x)\phi_N(x)v(x)\,dx \tag{16}$$

[6] This also assures that $\deg(H_{N-1}) = N-1$.

[7] You might want to review your calculus notes ☺

[8] The limit is not too difficult to calculate, try it! If you get stuck, see appendix B.

and substitute (8) in the first term

$$\int_a^b Q_{N-1}(x)v(x)\,dx = \int_a^b \sum_{j=1}^{N} q_{2N-1}(x_j)\frac{\phi_N(x)}{\phi'_N(x_j)(x-x_j)}v(x)\,dx\,. \tag{17}$$

Expanding r in the orthogonal basis $\{\phi\}$

$$r(x) = \sum_{k=0}^{N-1}\beta_k\phi_k(x) \tag{18}$$

and applying property (4) to the second term in (16)

$$\int_a^b r(x)\phi_N(x)v(x)\,dx = \sum_{k=0}^{N-1}\beta_k\int_a^b \phi_N(x)\phi_k(x)v(x)\,dx = \sum_{k=0}^{N-1}\beta_k A_k^2\delta_{k,N} = 0\,. \tag{19}$$

Equating to the RHS in (15)

$$\sum_{j=1}^{N} q_{2N-1}(x_j)\int_a^b \frac{\phi_N(x)v(x)}{\phi'_N(x_j)(x-x_j)}\,dx = \sum_{i=1}^{N} w_i q_{2N-1}(x_i)\,. \tag{20}$$

From this last expression we find the weights[9], calculating N integrals, one for each root x_i.

Second result: We'll calculate our weights with the formula $w_i = \dfrac{1}{\phi'_N(x_i)}\displaystyle\int_a^b \frac{\phi_N(x)v(x)}{x-x_i}\,dx$.

2.3 Mathemagics

Virtually we've solved our problem of finding $\{x_i\}$ and $\{w_i\}$, but we've gone from calculating one single integral to calculating N integrals, one for each of the weights. This is rather counter-intuitive and time-consuming, specially if we choose a large value for N in order to obtain a *good* approximation.

We start solving this problem with something quite simple, adding and subtracting

$$\frac{1}{x-x_i} = \frac{1}{x-x_i} + \frac{1}{x-x_i}\left(\frac{x}{x_i}\right)^k - \frac{1}{x-x_i}\left(\frac{x}{x_i}\right)^k \tag{21}$$

$$= \frac{1}{x-x_i}\left[1-\left(\frac{x}{x_i}\right)^k\right] + \frac{1}{x-x_i}\left(\frac{x}{x_i}\right)^k \tag{22}$$

$$= \frac{1}{x_i^k}\frac{x_i^k - x^k}{x-x_i} + \frac{1}{x_i^k}\frac{x^k}{x-x_i}\,. \tag{23}$$

Using equality (23) we calculate the integral

$$\int_a^b \frac{\phi_N(x)v(x)}{x-x_i}\,dx = \frac{1}{x_i^k}\int_a^b \phi_N(x)v(x)\frac{x_i^k-x^k}{x-x_i}\,dx + \frac{1}{x_i^k}\int_a^b \frac{x^k\phi_N(x)v(x)}{x-x_i}\,dx\,. \tag{24}$$

Note that the fraction $(x_i^k - x^k)/(x-x_i)$ in the first term is a polynomial of degree $k-1$ (both numerator and denominator share a common root, x_i). If $k \le N$, this polynomial can be expanded in the interval $[a,b]$ as

$$\frac{x_i^k - x^k}{x-x_i} = \sum_{m=0}^{N-1}\gamma_m\phi_m(x)\,.$$

[9] Note that both i and j are dummy indices.

Applying property (4) in the first term of (24)

$$\frac{1}{x_i^k}\int_a^b \phi_N(x)v(x)\frac{x_i^k-x^k}{x-x_i}\,dx=\sum_{m=0}^{N-1}\frac{\gamma_m}{x_i^k}\int_a^b \phi_N(x)\phi_m(x)v(x)\,dx=\sum_{m=0}^{N-1}\frac{\gamma_m}{x_i^k}A_m^2\delta_{m,N}=0\,, \tag{25}$$

and (24) reduces to

$$\int_a^b \frac{\phi_N(x)v(x)}{x-x_i}\,dx=\frac{1}{x_i^k}\int_a^b \frac{x^k\phi_N(x)v(x)}{x-x_i}\,dx\,. \tag{26}$$

This “decomposition" is generalizable to any polynomial S with $\deg(S)\le N$

$$\frac{1}{x-x_i}=\frac{1}{x-x_i}\left[1-\frac{S(x)}{S(x_i)}\right]+\frac{1}{x-x_i}\frac{S(x)}{S(x_i)} \tag{27}$$

$$\int_a^b \frac{\phi_N(x)v(x)}{x-x_i}\,dx=\frac{1}{S(x_i)}\int_a^b \frac{S(x)\phi_N(x)v(x)}{x-x_i}\,dx\,. \tag{28}$$

In particular, we choose $S=\phi_{N-1}$ and rewrite the expression for our weights

$$w_i=\frac{1}{\phi_N'(x_i)\phi_{N-1}(x_i)}\int_a^b \frac{\phi_N(x)\phi_{N-1}(x)v(x)}{x-x_i}\,dx\,. \tag{29}$$

Remember from (9) that $\phi_N(x)=c_N(x-x_1)(x-x_2)\cdots(x-x_i)\cdots(x-x_N)$ and note that $\phi_N(x)/(x-x_i)$ is a polynomial of degree $N-1$, which can be written as[10]

$$\frac{\phi_N(x)}{x-x_i}=c_Nx^{N-1}+\sum_{k=0}^{N-2}C_kx^k=c_Nx^{N-1}+\sum_{k=0}^{N-2}\epsilon_k\phi_k(x)\,. \tag{30}$$

Substituting in the integral of (29) and applying property (4)

$$\int_a^b \frac{\phi_N(x)\phi_{N-1}(x)v(x)}{x-x_i}\,dx=c_N\int_a^b x^{N-1}\phi_{N-1}(x)v(x)\,dx+\sum_{k=0}^{N-2}\epsilon_k\int_a^b \phi_k(x)\phi_{N-1}(x)v(x)\,dx \tag{31}$$

$$=c_N\int_a^b x^{N-1}\phi_{N-1}(x)v(x)\,dx+\sum_{k=0}^{N-2}\epsilon_kA_k^2\delta_{k,N-1} \tag{32}$$

$$=c_N\int_a^b x^{N-1}\phi_{N-1}(x)v(x)\,dx\,. \tag{33}$$

Finally, to get rid of the x^{N-1} in (33) we write

$$\phi_{N-1}(x)=c_{N-1}x^{N-1}+\sum_{k=0}^{N-2}D_kx^k=c_{N-1}x^{N-1}+\sum_{k=0}^{N-2}\zeta_k\phi_k \tag{34}$$

$$x^{N-1}=\frac{\phi_{N-1}(x)}{c_{N-1}}-\sum_{k=0}^{N-2}\frac{\zeta_k}{c_{N-1}}\phi_k\,. \tag{35}$$

Substituting in the integral of (33) and applying property (4)

$$c_N\int_a^b x^{N-1}\phi_{N-1}(x)v(x)\,dx=\frac{c_N}{c_{N-1}}\int_a^b \phi_{N-1}^2(x)v(x)\,dx-\sum_{k=0}^{N-2}\frac{c_N\zeta_k}{c_{N-1}}\int_a^b \phi_k(x)\phi_{N-1}(x)v(x)\,dx \tag{36}$$

$$=\frac{c_N}{c_{N-1}}A_{N-1}^2-\sum_{k=0}^{N-2}\frac{c_N\zeta_k}{c_{N-1}}\delta_{k,N-1} \tag{37}$$

$$=\frac{c_N}{c_{N-1}}A_{N-1}^2\,. \tag{38}$$

[10] Once again, we can expand the polynomial $\sum_{k=0}^{N-2}C_kx^k$ in the orthogonal basis $\{\phi\}$.

Tracing these results back to (29) we obtain a very compact expression for the weights

$$w_i = \frac{A_{N-1}^2}{\phi'_N(x_i)\phi_{N-1}(x_i)} \frac{c_N}{c_{N-1}} \,. \tag{39}$$

Conclusion: Choosing an $L_v^2[a,b]$ orthogonal polynomial basis $\{\phi\}$, our integration nodes will be the roots of ϕ_N, $\{x_i \,|\, \phi_N(x_i) = 0\}$, and we'll calculate the weights using the formula $w_i = \frac{A_{N-1}^2}{\phi'_N(x_i)\phi_{N-1}(x_i)} \frac{c_N}{c_{N-1}}$, with A^2 the normalization constant and c the coefficient of the largest monomial.

3 Gauss-Legendre quadrature

In the previous section we've derived the general form of Gaussian quadratures. Let's remember the relevant formulae

$$\int_a^b g(x)v(x)\,dx \simeq \sum_{i=1}^{N} w_i g(x_i) \tag{40}$$

$$x_i \to \phi_N(x_i) = 0 \tag{41}$$

$$w_i = \frac{A_{N-1}^2}{\phi_N'(x_i)\phi_{N-1}(x_i)}\frac{c_N}{c_{N-1}}\ . \tag{42}$$

Note that the only variability lies in the orthogonal polynomial basis $\{\phi\}$. The question that arrises is: what basis do we use? The choice depends on the interval $[a, b]$.

If it's semi-infinte ($a = 0, b = \infty$), an appropriate basis are Laguerre polynomials; if the interval is infinite ($a = -\infty, b = \infty$), we'd use Hermite polynomials. If the interval is finite we can use Legendre polynomials or Chebyshev polynomials of the second kind. We choose the former since their weight function is $v(x) = 1$, as opposed to $v(x) = \sqrt{1-x^2}$ for the latter.

3.1 Interval shift

Legendre polynomials are defined in the interval $x \in [-1, 1]$, i.e. the conclusions of the previous section are valid *only* if $a = -1$ and $b = 1$. If we want to compute an integral with other bounds

$$I = \int_a^b f(y)\,dy \tag{43}$$

we have to perform a change of variables

$$y = \frac{b-a}{2}x + \frac{a+b}{2} \to dy = \frac{b-a}{2}dx \tag{44}$$

$$I = \frac{b-a}{2}\int_{-1}^{1} f(x)\,dx\ . \tag{45}$$

3.2 Normalization constant

The normalization constant for Legendre polynomials is

$$A_n^2 = \int_{-1}^{1} P_n^2(x)\,dx = \frac{2}{2n+1}\ , \tag{46}$$

and particularly of our interest

$$A_{N-1}^2 = \frac{2}{2(N-1)+1} = \frac{2}{2N-1}\ . \tag{47}$$

3.3 Rodrigues' formula

Legendre polynomials may be expressed using Rodrigues' formula

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(\left(x^2-1\right)^n\right)\ , \tag{48}$$

which can be manipulated slightly

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(x^{2n} + O\left(x^{2n-1}\right)\right) = \frac{1}{2^n n!}\left(\frac{(2n)!}{n!}x^n + O\left(x^{n-1}\right)\right) . \tag{49}$$

From this expression we find

$$c_n = \frac{1}{2^n n!}\frac{(2n)!}{n!} = \frac{(2n)!}{2^n (n!)^2} , \tag{50}$$

and particularly

$$c_N = \frac{(2N)!}{2^N (N!)^2} \tag{51}$$

$$c_{N-1} = \frac{(2N-2)!}{2^{N-1}((N-1)!)^2} . \tag{52}$$

Now we can compute the fraction

$$\begin{aligned}
\frac{c_N}{c_{N-1}} &= \frac{(2N)!}{2^N (N!)^2}\frac{2^{N-1}((N-1)!)^2}{(2N-2)!} && (53)\\
&= \frac{2N\cdot(2N-1)\cdot(2N-2)!}{2\cdot 2^{N-1}\cdot N^2\cdot((N-1)!)^2}\frac{2^{N-1}((N-1)!)^2}{(2N-2)!} && (54)\\
&= \frac{2N\cdot(2N-1)}{2\cdot N^2} && (55)\\
&= \frac{2N-1}{N} . && (56)
\end{aligned}$$

3.4 Recurrence relations

There are a few interesting recurrence relations involving Legendre polynomials. Of particular interest are

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{57}$$

and

$$(x^2-1)P_n'(x) = nxP_n(x) - nP_{n-1}(x) . \tag{58}$$

We'll use the first in our algorithm and from the second

$$(x_i^2-1)P_N'(x_i) = Nx_iP_N(x_i) - NP_{N-1}(x_i) \rightarrow P_N'(x_i) = \frac{NP_{N-1}(x_i)}{1-x_i^2} , \tag{59}$$

where we've used $P_N(x_i) = 0$.

Plugging (47), (56) and (59) into (42) we find

$$w_i = \frac{2}{N^2}\frac{1-x_i^2}{P_{N-1}^2(x_i)} . \tag{60}$$

4 Computational algorithm

User input parameters:

- number of integration nodes, N
- integral bounds, a and b
- function, $f(y)$

The steps we have to implement are:

1. find the N roots of P_N, $\{x_i\}$
2. calculate the weights, $\{w_i\}$
3. perform the sum $\sum_{i=1}^{N} w_i f(x_i)$

To find the N roots of P_N we'll combine the secant and bisection methods. In order to do so, we'll need to narrow the interval where each root lies.

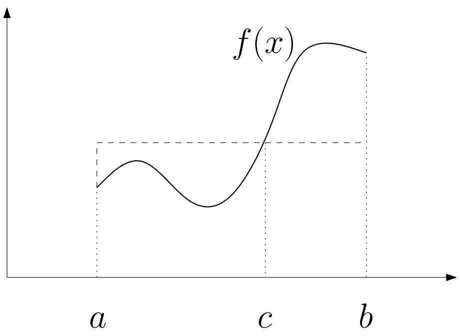
1. start with a step of $\Delta = 2/N$
2. sweep the interval $[-1, 1]$ with step Δ, $x_{j+1} = x_j + \Delta$
   - compare the sign of $P_N(x_j)$ and $P_N(x_{j+1})$
   - if there's a change of sign, record x_j and x_{j+1}
3. count the number of records, M
   - if $M < N$ repeat step 2 with $\Delta = \Delta/10$
   - if $M = N$ we're finished

Now that we've delimited the interval where each root lies we apply the enhanced secant method to find the integration nodes with a fixed precision, ϵ. This enhanced method consists simply in correcting the secant result, $x_{i+1} = x_i - f(x_i)(x_i - x_{i-1})/(f(x_i) - f(x_{i-1}))$, if it diverges from the interval, using the bisection result instead.

To compute the values of the Lagrange polynomials we'll need the recurrence relation (57) and the fact that $P_0 = 1$ and $P_1 = x$.

Remember that we'll have to remap the variable x into the variable y, using (44), in order to compute $f(x_i)$.

Appendix A



Given a well-behaved function f, there exists a value c, $a < c < b$, such that

$$\int_a^b f(x)\, dx = (b-a) \cdot f(c) \; . \tag{61}$$

Graphically this means that the area below the curve (between a and c) has the same vale as the area above the curve (between c and b).

Appendix B

We want to show

$$\lim_{x \to x_k} \frac{\phi_N(x)}{\phi'_N(x_j)(x-x_j)} = \delta_{j,k} \; . \tag{62}$$

Rewriting (9)

$$\phi_N(x) = c_N(x-x_1)(x-x_2)\cdots(x-x_N) = c_N \prod_{i=1}^{N}(x-x_i) \; . \tag{63}$$

The denominator $x - x_j$ cancels with a term in the numerator, thus

$$\frac{\phi_N(x)}{x-x_j} = c_N \prod_{i \neq j}^{N}(x-x_i) \; . \tag{64}$$

Differentiating (63)

$$\begin{aligned}\phi'_N(x) = c_N\, [(x-x_2)(x-x_3)\cdots(x-x_N) + (x-x_1)(x-x_3)\cdots(x-x_N) + \cdots \\ + (x-x_1)(x-x_2)\cdots(x-x_{N-1})] = c_N \sum_{m=1}^{N} \frac{\prod_{i=1}^{N}(x-x_i)}{x-x_m} = c_N \sum_{m=1}^{N} \prod_{i \neq m}^{N}(x-x_i) \; .\end{aligned} \tag{65}$$

Once again, the denominator $x - x_m$ cancels with a term in the numerator. Considering a specific value of $x = x_j$

$$\phi'_N(x_j) = c_N \sum_{m=1}^{N} \prod_{i \neq m}^{N}(x_j - x_i) \; . \tag{66}$$

Most of the summands in (66) include the term $i = j \to x_j - x_i = 0$. The only summand that survives is $i \neq j$, i.e. $m = j$ and

$$\phi'_N(x_j) = c_N \prod_{i \neq j} (x_j - x_i) \; . \tag{67}$$

Substituting these expressions in the limit

$$\lim_{x \to x_k} \frac{\phi_N(x)}{\phi'_N(x_j)(x - x_j)} = \lim_{x \to x_k} \frac{c_N \prod_{i \neq j}(x - x_i)}{c_N \prod_{i \neq j}(x_j - x_i)} = \frac{\prod_{i \neq j}(x_k - x_i)}{\prod_{i \neq j}(x_j - x_i)} \; . \tag{68}$$

It's clear that in the case $k = j$ numerator and denominator are equal and the limit's value is 1. On the other hand, if $k \neq j$ there's a term in the numerator with $i = k \to x_k - x_i = 0$ and the limit's value is 0. Therefore

$$\lim_{x \to x_k} \frac{\phi_N(x)}{\phi'_N(x_j)(x - x_j)} = \delta_{j,k} \qquad \text{Q.E.D.} \tag{69}$$