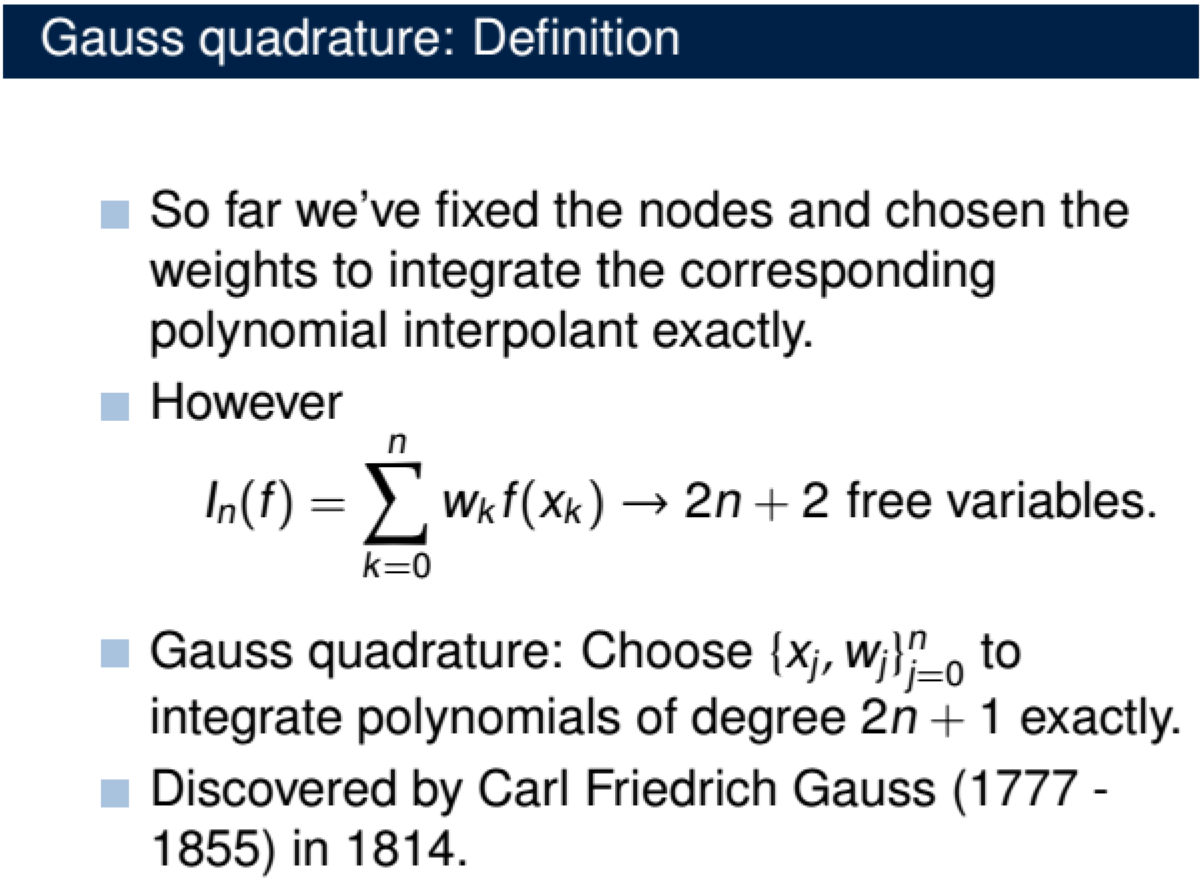
LECTURE 8: GAUSE QUADRATURE

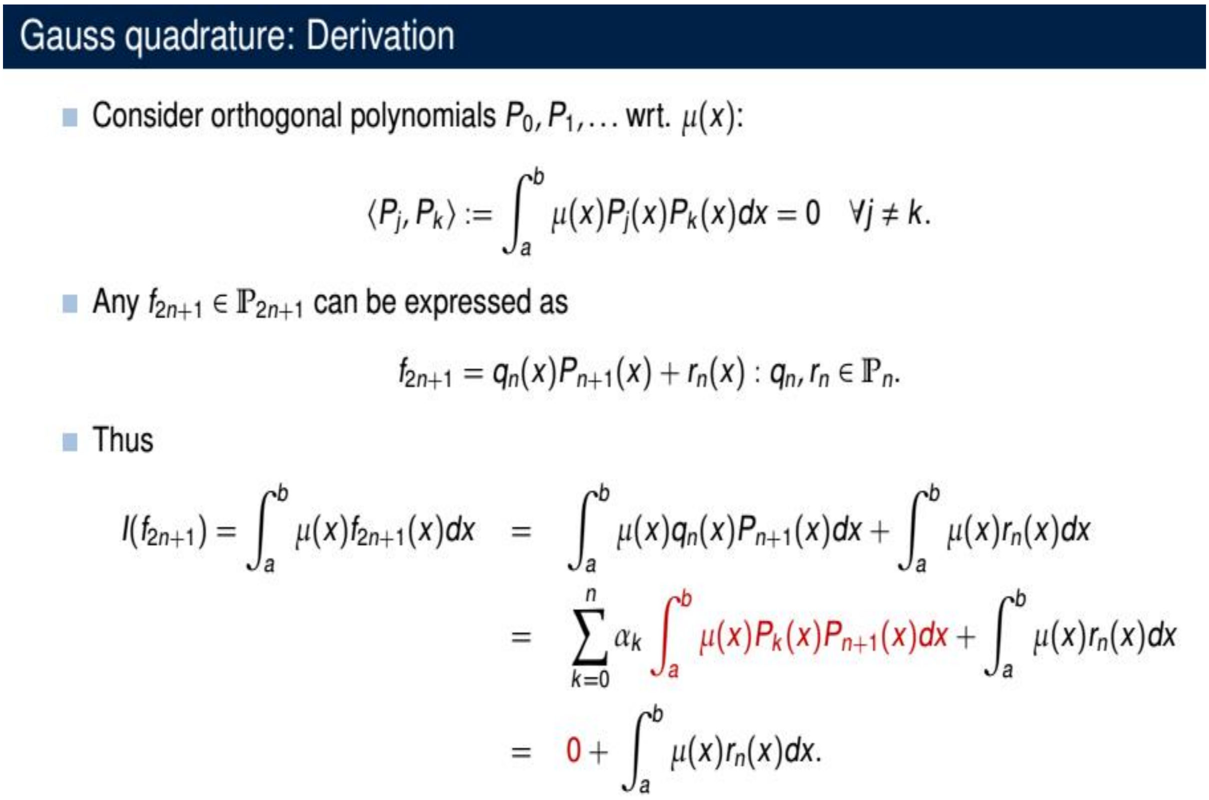


Gauss quadrature: Definition

- So far we've fixed the nodes and chosen the weights to integrate the corresponding polynomial interpolant exactly.
- However

$$I_n(f) = \sum_{k=0}^{n} w_k f(x_k) \to 2n + 2 \text{ free variables.}$$

- Gauss quadrature: Choose $\{x_j, w_j\}_{j=0}^{n}$ to integrate polynomials of degree $2n + 1$ exactly.
- Discovered by Carl Friedrich Gauss (1777 - 1855) in 1814.



Gauss quadrature: Derivation

- Consider orthogonal polynomials $P_0, P_1, \ldots$ wrt. $\mu(x)$:

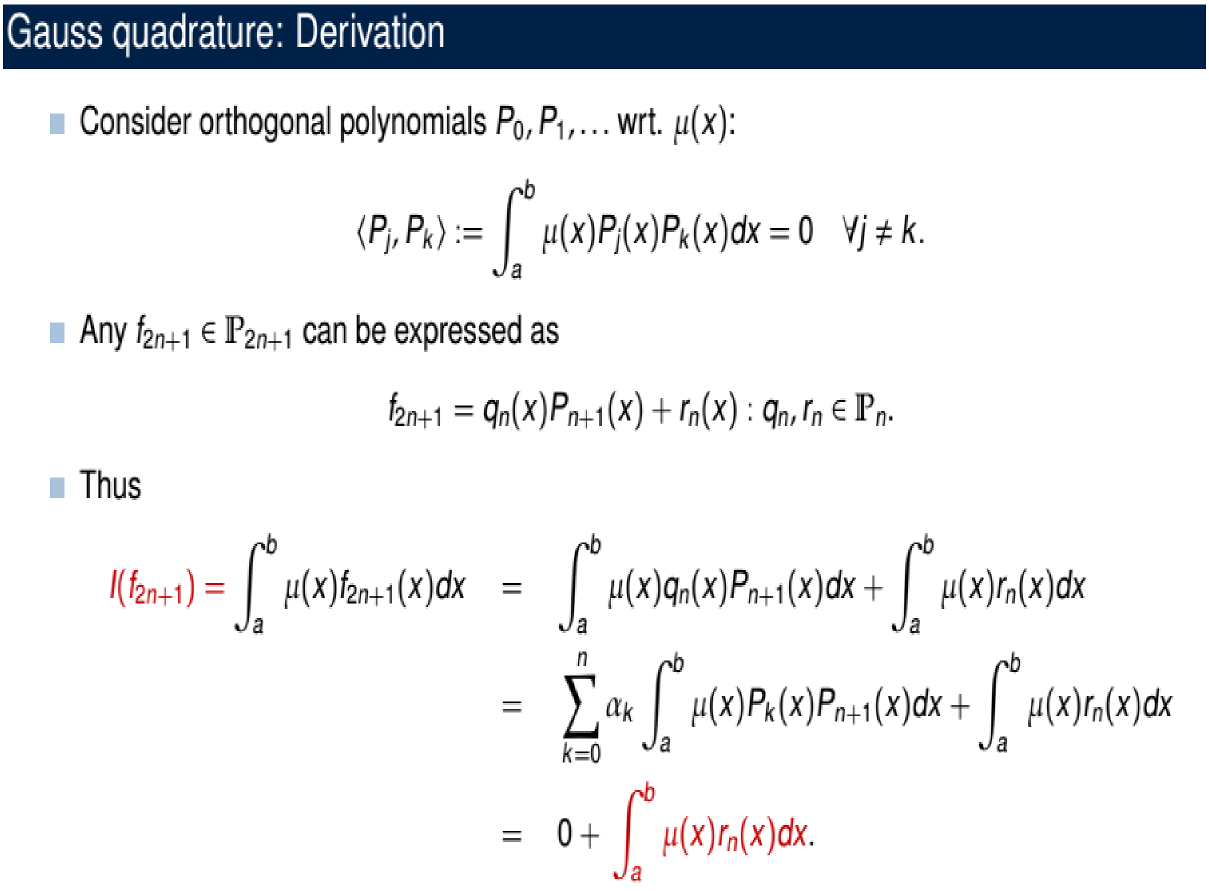
$$\langle P_j, P_k \rangle := \int_a^b \mu(x) P_j(x) P_k(x) dx = 0 \quad \forall j \neq k.$$

- Any $f_{2n+1} \in \mathbb{P}_{2n+1}$ can be expressed as

$$f_{2n+1} = q_n(x) P_{n+1}(x) + r_n(x) : q_n, r_n \in \mathbb{P}_n.$$

- Thus

$$\begin{aligned} I(f_{2n+1}) = \int_a^b \mu(x) f_{2n+1}(x) dx &= \int_a^b \mu(x) q_n(x) P_{n+1}(x) dx + \int_a^b \mu(x) r_n(x) dx \\ &= \sum_{k=0}^{n} \alpha_k \int_a^b \mu(x) P_k(x) P_{n+1}(x) dx + \int_a^b \mu(x) r_n(x) dx \\ &= 0 + \int_a^b \mu(x) r_n(x) dx. \end{aligned}$$



Gauss quadrature: Derivation

- Consider orthogonal polynomials $P_0, P_1, \ldots$ wrt. $\mu(x)$:

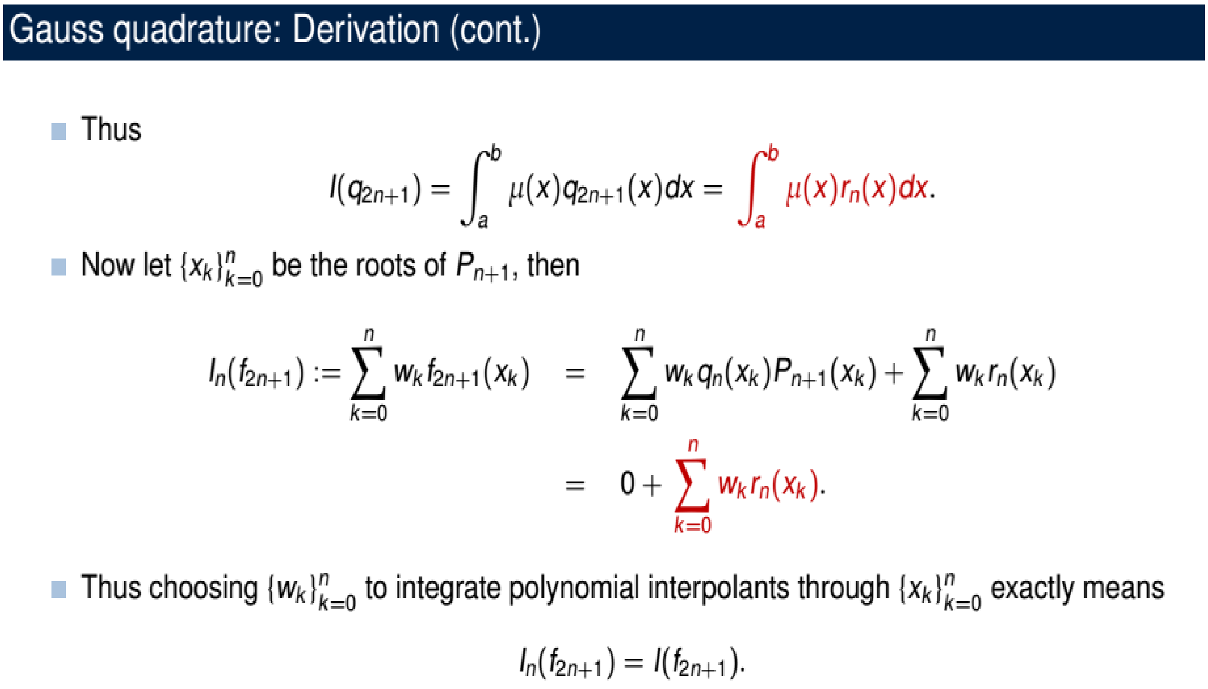
$$\langle P_j, P_k \rangle := \int_a^b \mu(x) P_j(x) P_k(x) dx = 0 \quad \forall j \neq k.$$

- Any $f_{2n+1} \in \mathbb{P}_{2n+1}$ can be expressed as

$$f_{2n+1} = q_n(x) P_{n+1}(x) + r_n(x) : q_n, r_n \in \mathbb{P}_n.$$

- Thus

$$\begin{aligned} I(f_{2n+1}) = \int_a^b \mu(x) f_{2n+1}(x) dx &= \int_a^b \mu(x) q_n(x) P_{n+1}(x) dx + \int_a^b \mu(x) r_n(x) dx \\ &= \sum_{k=0}^{n} \alpha_k \int_a^b \mu(x) P_k(x) P_{n+1}(x) dx + \int_a^b \mu(x) r_n(x) dx \\ &= 0 + \int_a^b \mu(x) r_n(x) dx. \end{aligned}$$



Gauss quadrature: Derivation (cont.)

- Thus

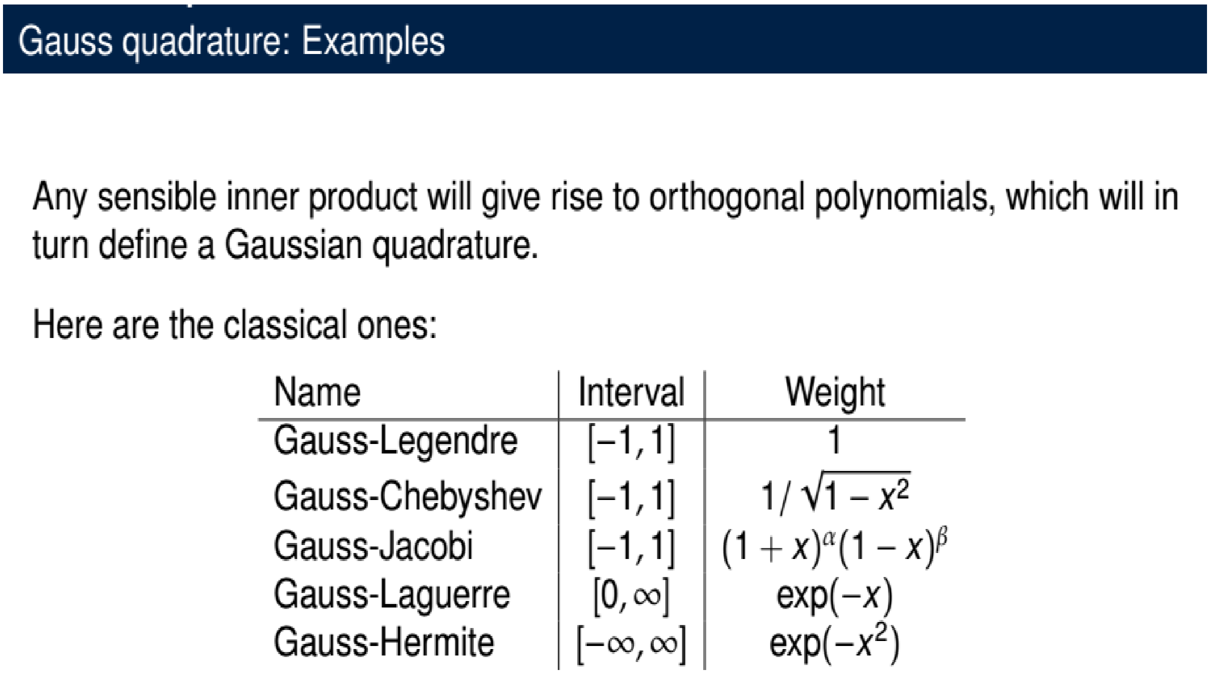
$$I(q_{2n+1}) = \int_a^b \mu(x) q_{2n+1}(x)dx = \int_a^b \mu(x) r_n(x)dx.$$

- Now let $\{x_k\}_{k=0}^n$ be the roots of P_{n+1}, then

$$\begin{aligned} I_n(f_{2n+1}) := \sum_{k=0}^{n} w_k f_{2n+1}(x_k) &= \sum_{k=0}^{n} w_k q_n(x_k) P_{n+1}(x_k) + \sum_{k=0}^{n} w_k r_n(x_k) \\ &= 0 + \sum_{k=0}^{n} w_k r_n(x_k). \end{aligned}$$

- Thus choosing $\{w_k\}_{k=0}^n$ to integrate polynomial interpolants through $\{x_k\}_{k=0}^n$ exactly means

$$I_n(f_{2n+1}) = I(f_{2n+1}).$$

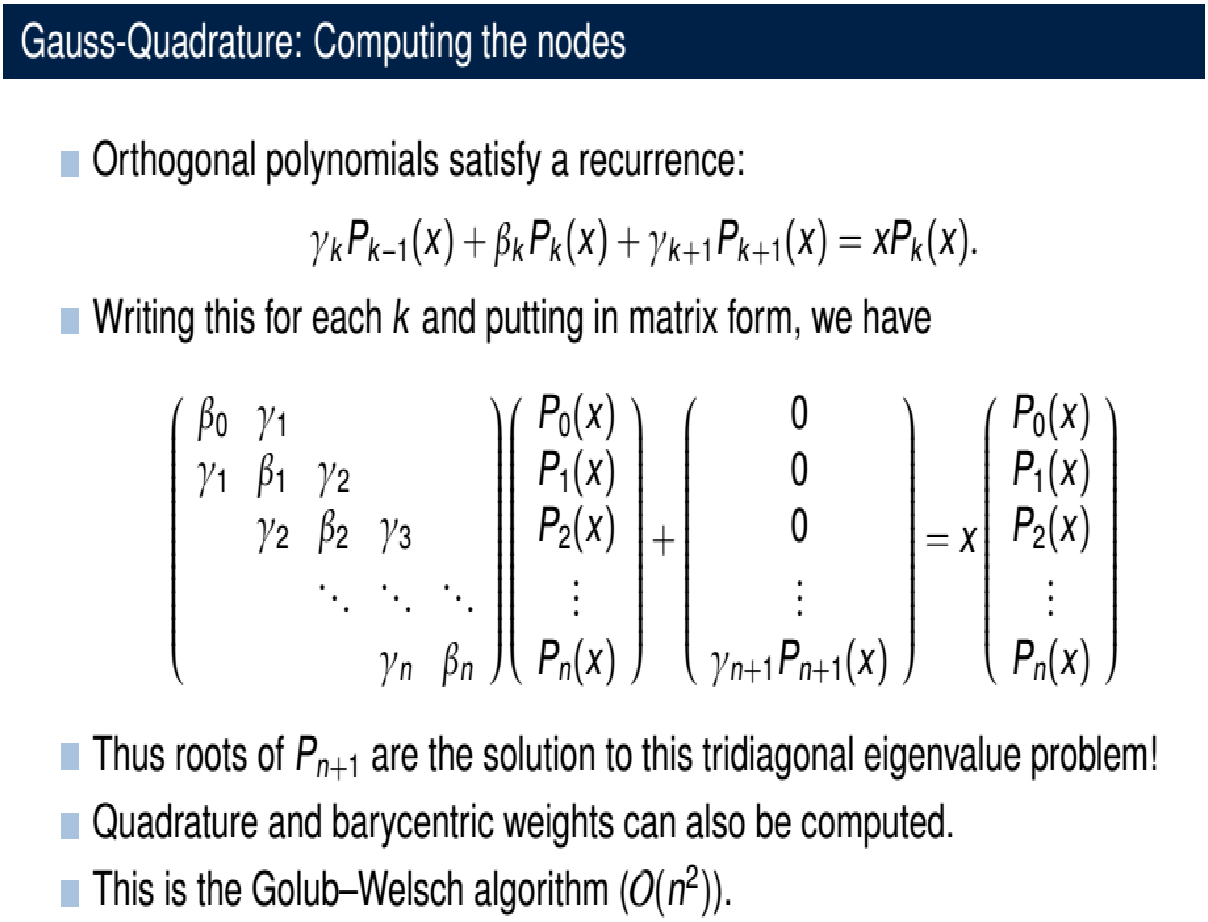


Gauss quadrature: Examples

Any sensible inner product will give rise to orthogonal polynomials, which will in turn define a Gaussian quadrature.

Here are the classical ones:

Name	Interval	Weight
Gauss-Legendre	$[-1,1]$	1
Gauss-Chebyshev	$[-1,1]$	$1/\sqrt{1-x^2}$
Gauss-Jacobi	$[-1,1]$	$(1+x)^{\alpha}(1-x)^{\beta}$
Gauss-Laguerre	$[0,\infty]$	$\exp(-x)$
Gauss-Hermite	$[-\infty,\infty]$	$\exp(-x^2)$



Gauss-Quadrature: Computing the nodes

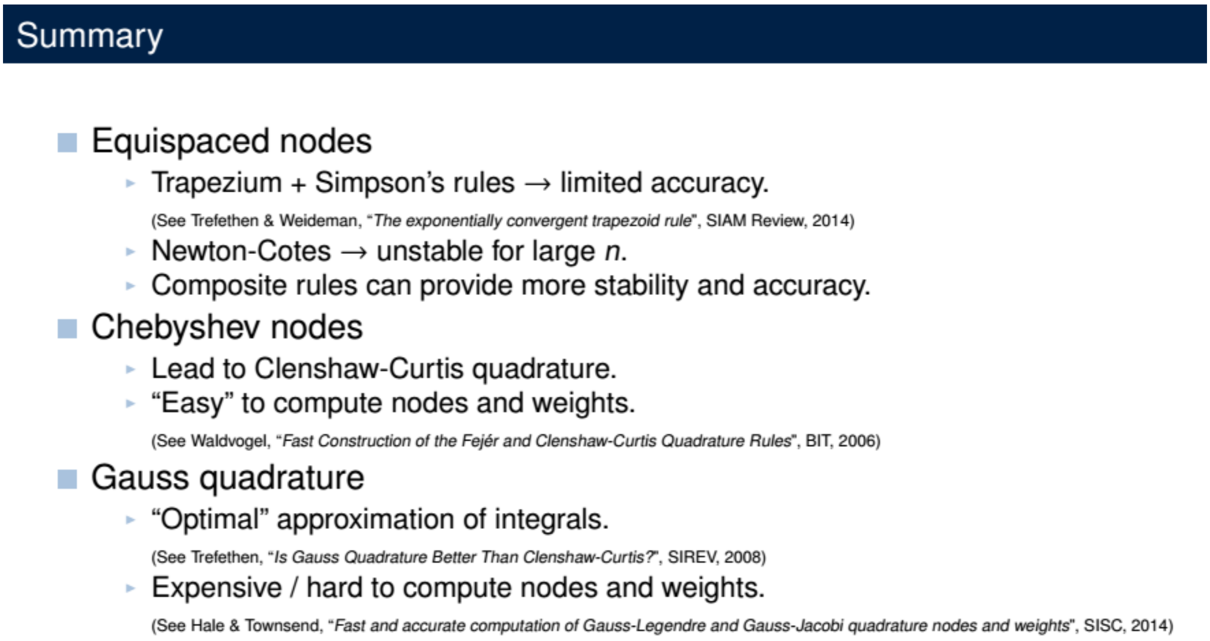
- Orthogonal polynomials satisfy a recurrence:

$$\gamma_k P_{k-1}(x) + \beta_k P_k(x) + \gamma_{k+1} P_{k+1}(x) = x P_k(x).$$

- Writing this for each k and putting in matrix form, we have

$$\begin{pmatrix} \beta_0 & \gamma_1 & & & \\ \gamma_1 & \beta_1 & \gamma_2 & & \\ & \gamma_2 & \beta_2 & \gamma_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \gamma_n & \beta_n \end{pmatrix} \begin{pmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ \vdots \\ P_n(x) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \gamma_{n+1} P_{n+1}(x) \end{pmatrix} = x \begin{pmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ \vdots \\ P_n(x) \end{pmatrix}$$

- Thus roots of P_{n+1} are the solution to this tridiagonal eigenvalue problem!
- Quadrature and barycentric weights can also be computed.
- This is the Golub–Welsch algorithm ($\mathcal{O}(n^2)$).



Summary

- Equispaced nodes
  - Trapezium + Simpson's rules → limited accuracy.
    (See Trefethen & Weideman, "*The exponentially convergent trapezoid rule*", SIAM Review, 2014)
  - Newton-Cotes → unstable for large n.
  - Composite rules can provide more stability and accuracy.
- Chebyshev nodes
  - Lead to Clenshaw-Curtis quadrature.
  - "Easy" to compute nodes and weights.
    (See Waldvogel, "*Fast Construction of the Fejér and Clenshaw-Curtis Quadrature Rules*", BIT, 2006)
- Gauss quadrature
  - "Optimal" approximation of integrals.
    (See Trefethen, "*Is Gauss Quadrature Better Than Clenshaw-Curtis?*", SIREV, 2008)
  - Expensive / hard to compute nodes and weights.
    (See Hale & Townsend, "*Fast and accurate computation of Gauss-Legendre and Gauss-Jacobi quadrature nodes and weights*", SISC, 2014)