Computing Gauss-Jacobi quadrature rules

Alex Townsend

Joint work with Nick Hale
NA Internal Seminar
Trinity Term 2012

An n-point Gauss–Jacobi quadrature rule:

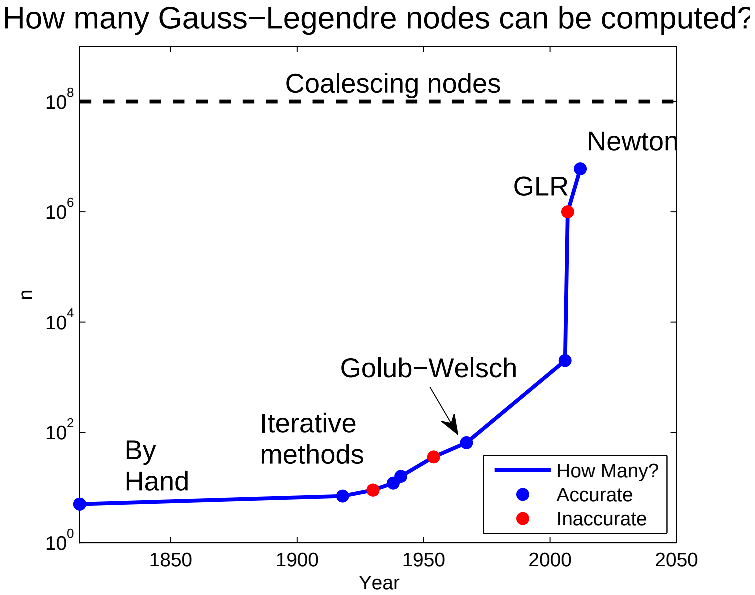
$$\int_{-1}^{1} w(x)f(x)dx \approx \sum_{k=1}^{n} w_k f(x_k)$$

with $w(x) = (1-x)^\alpha(1+x)^\beta$, $\alpha, \beta > -1$.

$x_k =$ simple roots of $P_n^{(\alpha,\beta)}$.

$$w_k = \frac{C_{n,\alpha,\beta}}{(1-x_k^2)\left[P_n^{(\alpha,\beta)\prime}(x_k)\right]^2}, \quad k = 1, \ldots, n.$$

Many formulae for the weights: Swarztrauber (2002), Yakimiv (1996).


How many Gauss−Legendre nodes can be computed?
Coalescing nodes
Newton
GLR
Golub−Welsch
Iterative
methods
By
Hand
How Many?
Accurate
Inaccurate
n
10^8
10^6
10^4
10^2
10^0
1850
1900
1950
2000
2050
Year

A set of Jacobi polynomials satisfy a 3-term recurrence

$$A_n P_n(x) = (x - B_n) P_{n-1}(x) - C_n P_{n-2}(x).$$

The zeros of $P_n(x)$ are the eigenvalues of the tridiagonal matrix

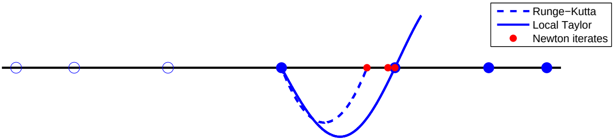
$$\begin{pmatrix} B_n & C_n & & & & \\ A_{n-1} & B_{n-1} & C_{n-1} & & & \\ & A_{n-2} & B_{n-2} & C_{n-2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_2 & B_2 & C_2 \\ & & & & A_1 & B_1 \end{pmatrix}.$$

It is a comrade matrix and hence, a strong linearisation (Mackay et al, 2007). It can be made symmetric tridiagonal.
Requires $\mathcal{O}(n^2)$ operations for nodes and weights. Best not to compute weights from eigenvectors.

A set of Jacobi polynomials satisfy a second-order differential equation

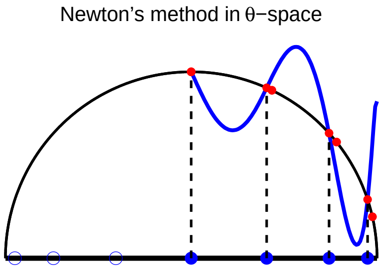
$$(1-x^2)P_n''(x) + a_n(x)P_n'(x) + b_n(x)P_n(x) = 0.$$

Predictor-corrector method:
Once at a root, step to the next by *predicting* with Runge–Kutta and then *correcting* with local Taylor approximations and Newton.



Requires $\mathcal{O}(n)$ operations for nodes and weights.

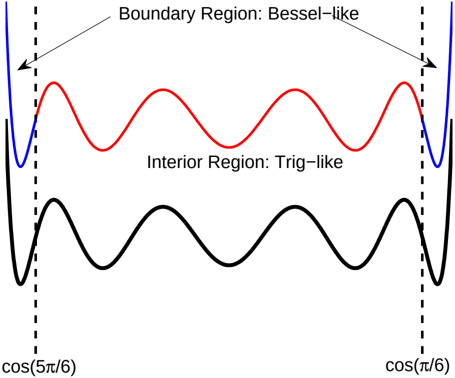
Newton's method in θ-space, $x = \cos(\theta)$.



We will need:

1. Fast and accurate Jacobi polynomial evaluation $\rightarrow$ asymptotic approximations.
2. Fast and accurate evaluation of derivative $\rightarrow$ recurrence relations.
3. Sufficiently good initial guesses $\rightarrow$ asymptotic approximations.

What does a Legendre polynomial look like?

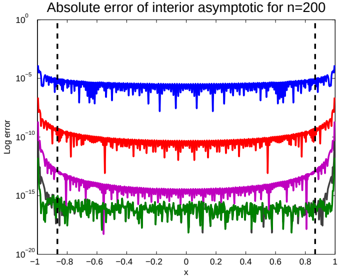


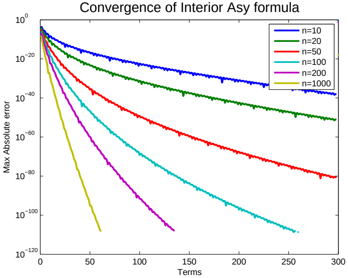
We use two asymptotic expansions: interior and boundary.

$$P_n(\cos\theta) \sim 2(-1)^n \sum_{m=0}^{M-1} \binom{-\frac{1}{2}}{m} \binom{m-\frac{1}{2}}{n} \frac{\cos(\alpha_{n,m})}{(2\sin\theta)^{m+\frac{1}{2}}} + R_{n,M}$$

where $\alpha_{n,m} = \left(n+m+\frac{1}{2}\right)\theta - \left(m+\frac{1}{2}\right)\frac{\pi}{2}$ and

$$|R_{n,M}| \le \frac{C}{(2\sin\theta)^{M+\frac{1}{2}}\, n^{M+\frac{1}{2}}}.$$



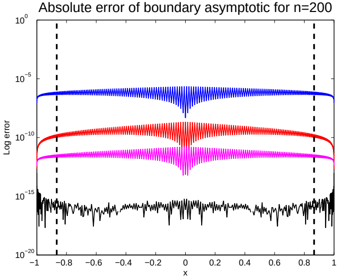


Boundary Asymptotic Formulae

$$P_n(\cos\theta) \sim \sqrt{\frac{\theta}{\sin(\theta)}}\left(J_0(\rho\theta)\sum_{m=0}^{M}\frac{A_m(\theta)}{\rho^{2m}} + \theta J_1(\rho\theta)\sum_{m=0}^{M-1}\frac{B_m(\theta)}{\rho^{2m+1}}\right),$$

where $\rho = n + \frac{1}{2}$. Only the first few terms are known explictly.

$$\text{error} = \begin{cases} \theta\mathcal{O}\left(n^{-2M-\frac{3}{2}}\right) & \frac{c}{n} \leq \theta \leq \frac{\pi}{2} \\ \theta^3\mathcal{O}\left(n^{-2M+\frac{1}{2}}\right) & 0 \leq \theta \leq \frac{c}{n}. \end{cases}$$



A recurrence relation for the derivative of P_n is

$$(1 - x^2)P_n'(x) = -nxP_n(x) + nP_{n-1}(x),$$

or in θ-space

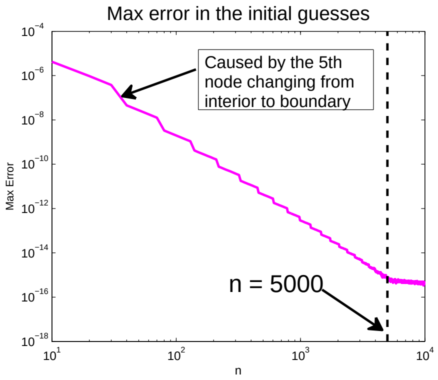
$$\sin(\theta)\frac{d}{d\theta}P_n(\cos(\theta)) = -n\cos(\theta)P_n(\cos(\theta)) + nP_{n-1}(\cos(\theta)).$$

Many quantities can be reused for derivative evaluation.

Initial Guesses

Sufficiently good = quadratically clustering near endpoints. (Petras, 1998)

Initial guesses come from the asymptotic formulae. Analogously, there are interior and boundary initial guesses. (Lether and Wenston, 1995).



Evaluating $J_0(\rho\theta)$ near Legendre nodes

Inbuilt bessel evaluation only gets 15-digits of absolute accuracy, we need one bit more.

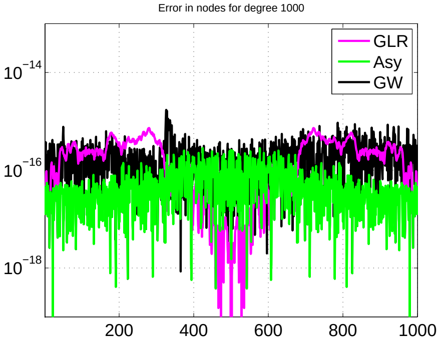
Instead, we evaluate the asymptotic expansion:

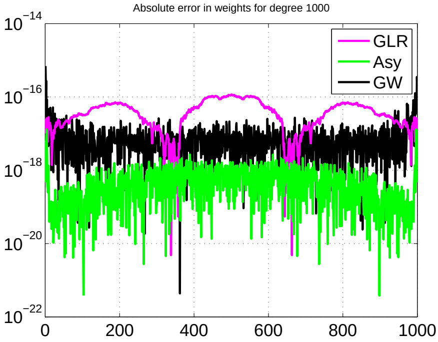
$$J_0(\rho\theta) \sim \left(\frac{2}{\pi\rho\theta}\right)^{\frac{1}{2}} \left(\cos\omega \sum_{k=0}^{\infty}(-1)^k \frac{a_{2k}}{(\rho\theta)^{2k}} - \sin\omega \sum_{k=0}^{\infty}(-1)^k \frac{a_{2k+1}}{(\rho\theta)^{2k+1}}\right) \quad (1)$$

where $\omega = \rho\theta - \frac{1}{4}\pi$ and do the argument reduction by ourselves.

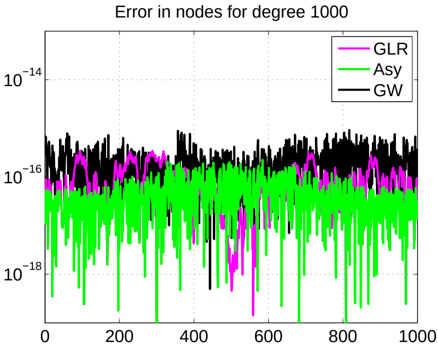
	Relative error	Absolute error
Inbuilt	2.54×10^{-9}	2.72×10^{-15}
Formula (1)	2.05×10^{-10}	4.30×10^{-17}

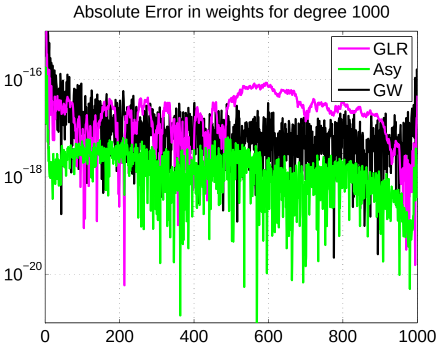
Gauss–Legendre nodes and weights ($\alpha = 0$, $\beta = 0$).





Gauss–Jacobi nodes and weights ($\alpha = 0.1$, $\beta = -0.3$).

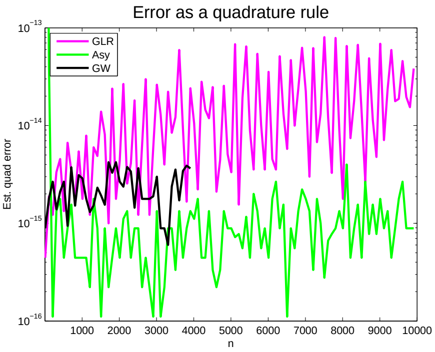


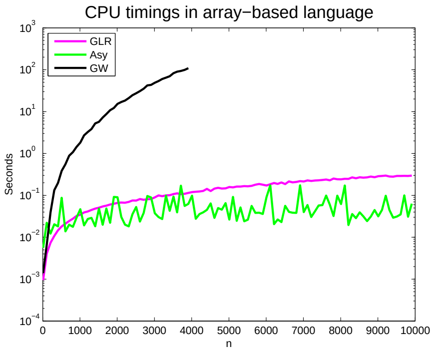


Error as a quadrature rule

Gauss quadrature integrates polynomials exactly. Approximate the error

$$\text{error} = \max_{j=0,\dots,10} \left| \int_{-1}^{1} w(x)x^j\,dx - \sum_{k=1}^{n} w_k x_k^j \right|.$$





Conclusions

1. Golub–Welsch method can be accurate, but has $\mathcal{O}(n^2)$ complexity.

2. GLR method is fast, but a little inaccurate for $n > 500$.

3. The method described here is fast and accurate for $n > 200$, but currently only implemented for Gauss–Jacobi nodes and weights.