Computing rational Gauss-Chebyshev quadrature formulas with complex poles

Karl Deckers Joris Van Deun Adhemar Bultheel

Department of Computer Science
K.U.Leuven

Augustus 7, 2006

Rational Gauss-Chebyshev quadrature

Algorithm to compute the nodes and weights for rational Gauss-Chebyshev quadrature formulas.

- Gauss quadrature:

$$\int_{-1}^{1} f(x)w(x)dx \approx \sum_{k=1}^{n} \lambda_{nk} f(x_{nk})$$

- Chebyshev weight functions:

$$w(x) = (1-x)^a(1+x)^b, \quad a, b \in \left\{\pm\frac{1}{2}\right\}$$

Notations

- $\overline{\mathbb{C}}$ Riemann sphere $\mathbb{C} \cup \{\infty\}$
- I interval $[-1, 1]$
- X^I complement of I with respect to a set X
- A_n sequence of poles $\{\alpha_1, \dots, \alpha_n\} \subset \overline{\mathbb{C}}^I$
- $\mathcal{L}_n$ space of rational functions with poles in A_n

Back to the quadrature formula

Theorem

There exist a set of nodes x_{nk} and weights λ_{nk}, $k = 1, \ldots, n$ so that the quadrature formula

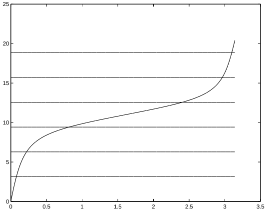
$$\int_{-1}^{1} f(x)w(x)dx \approx \sum_{k=1}^{n} \lambda_{nk} f(x_{nk})$$

is exact for $f \in \mathcal{L}_{n-1} \cdot \overline{\mathcal{L}_{n-1}}$. In the special case in which α_n is real, this quadrature formula is exact for $f \in \mathcal{L}_n \cdot \overline{\mathcal{L}_{n-1}}$.

Nodes and weights

nodes

- $x_{nk} = \cos\theta_{nk} \in I$ satisfy $F_n(\theta_{nk}) = \pi k$, $k = 1, \ldots, n$
- $F_n(\theta)$ is strictly increasing with increasing $\theta \in [0, \pi]$
- the nodes have to be computed numerically, e.g. using Newton's method



Nodes and weights

weights

- the weights are given by $\lambda_{nk} = G_n(x_{nk})$, $k = 1, \ldots, n$
- the weights can be computed straightforwardly

Computing the nodes

Two methods for determining a set of initial values for Newton's method:

- Asymptotic Zero Distribution (AZD)
- Asymptotic Inflection Point Distribution (AIPD)

Asymptotic zero distribution

Theorem

Assume the sequence of poles is bounded away from I *and the asymptotic distribution of the poles is given by a measure* ν *on (a subset of)* $\overline{\mathbb{C}}^I$*, then the asymptotic distribution of the nodes is given by an absolutely continuous measure* μ *and the density of the nodes on* $[-1, x]$ *is given by* $t(x) = \int_{-1}^{x} \mu'(u) du$*.*

Asymptotic zero distribution

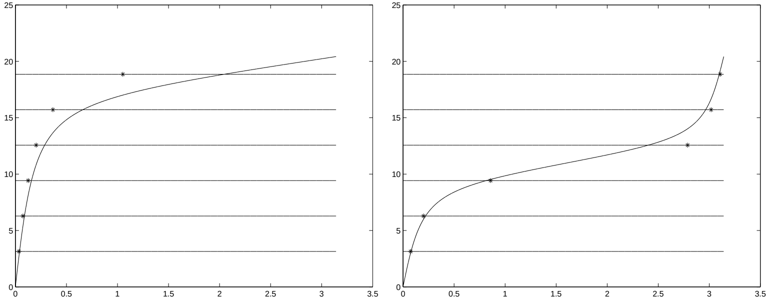
distribution of the poles is known

- $\lim_{n\to\infty} \alpha_n = \alpha \in \overline{\mathbb{R}}^I$
- $\theta_{n,k}^{(0)} = f_{AZD}(t_{n,k})$
- $f_{AZD}(t)$ is the inverse of $t(\theta)$
- $\{t_{n,k}\}_{k=1}^{n}$ is a set of n equally distributed points in $[0,1]$

distribution of the poles is unknown

- $t(\theta)$ can be approximated by a finite sum $t_n(\theta)$
- we can use the cubic interpolating spline $s_{AZD}(t)$ to approximate the inverse of $t_n(\theta)$
- $\theta_{n,k}^{(0)} = s_{AZD}(t_{n,k})$

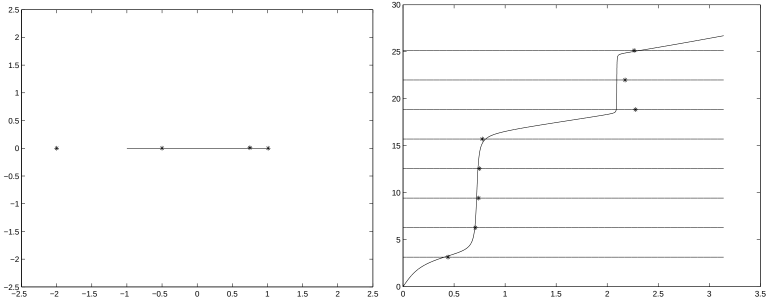
Asymptotic zero distribution



Asymptotic zero distribution

Problem : does not work well for poles close to the boundary introducing large local maxima of $\frac{dF_n(\theta)}{d\theta}$

```
Example: a = [-.5+i*1e-3*ones(1,2),.75+i*1e-2*ones(1,4),
                1.01,-2]
w = 2        %w(x) = \sqrt((1-x)/(1+x))
```



Asymptotic inflection point distribution

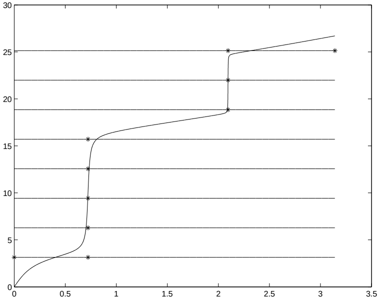
define

- $\theta_{b_j}^{(0)} = f_{AIPD}(\alpha_j) \approx \theta_{b_j}$
- $l_j = \frac{F_n\left(\theta_{b_j}^{(0)}\right)}{\pi}$

approximate from the left
$\theta_{n,k}^{(0)} = \theta_{b_j}^{(0)}$, with $j = \arg\max_j(l_j \leq k)$

approximate from the right
$\theta_{n,k}^{(0)} = \theta_{b_j}^{(0)}$, with $j = \arg\min_j(l_j \geq k)$

Asymptotic inflection point distribution



Numerical example

Syntax for `gqcorf`

```
[x,L,err,fail] = gqcorf(a,w)
```

where

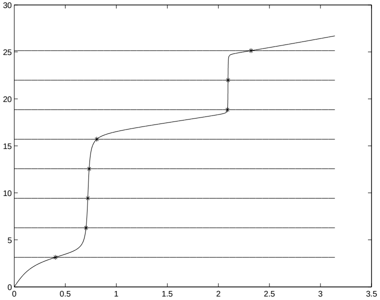
- `x` vector with the resulting nodes
- `L` vector with the resulting weights
- `err` (optional) to check whether the computations succeeded
- `fail` (optional) vector with indices of nodes/weights for which the computations failed (if any)
- `a` vector with poles
- `w` (optional) choice of weight function

Numerical example

```
Example: a = [-.5+i*1e-3*ones(1,2),.75+i*1e-2*ones(1,4),
                1.01,-2]
w = 2        %w(x) = \sqrt((1-x)/(1+x))
```

method	Δx	$\Delta\lambda$	$\lvert\pi - \sum_{k=1}^{n} \lambda_k\rvert$	i_{total}	found
bisection	0	0	1.6×10^{-14}	411	8
AZD	2.3×10^{-15}	2.0×10^{-14}	0	36	6
AIPD				17	2

Numerical example



Numerical example

The complexity of the algorithm is of order $\vartheta(m \times n)$. If $m << n$ the complexity is of order $\vartheta(n)$.
Example: $m = 5$

n	t	n	t	n	t
8	0.02	256	0.14	8192	3.17
16	0.01	512	0.28	16384	6.35
32	0.02	1024	0.55	32768	12.72
64	0.05	2048	1.08	65536	25.65
128	0.07	4096	2.01	131072	51.26

Numerical example

Example: $n = 8192$

m	t	m	t	m	t
5	3.78	65	5.80	1025	43.41
9	4.25	129	8.17	2049	82.47
17	4.13	257	11.57	4097	169.41
33	4.96	513	24.80	8192	344.23