Orthogonal Polynomials, Quadrature, and Approximation: Computational Methods and Software (in Matlab)

Walter Gautschi

wxg@cs.purdue.edu

Purdue University

Orthogonal Polynomials

Orthogonal Polynomials

- Recurrence coefficients

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods
- Cauchy integrals of orthogonal polynomials

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods
- Cauchy integrals of orthogonal polynomials
- Modification algorithms

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods
- Cauchy integrals of orthogonal polynomials
- Modification algorithms

Sobolev Orthogonal Polynomials

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods
- Cauchy integrals of orthogonal polynomials
- Modification algorithms

Sobolev Orthogonal Polynomials

- Moment-based algorithm

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods
- Cauchy integrals of orthogonal polynomials
- Modification algorithms

Sobolev Orthogonal Polynomials

- Moment-based algorithm
- Discretization algorithm

Orthogonal Polynomials

- Recurrence coefficients
- Modified Chebyshev algorithm
- Discrete Stieltjes and Lanczos algorithm
- Discretization methods
- Cauchy integrals of orthogonal polynomials
- Modification algorithms

Sobolev Orthogonal Polynomials

- Moment-based algorithm
- Discretization algorithm
- Zeros

Quadrature

Quadrature

- Gauss-type quadrature formulae

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula
- Gauss-Kronrod quadrature

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula
- Gauss-Kronrod quadrature
- Gauss-Turán quadrature

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula
- Gauss-Kronrod quadrature
- Gauss-Turán quadrature
- Quadrature formulae based on rational functions

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula
- Gauss-Kronrod quadrature
- Gauss-Turán quadrature
- Quadrature formulae based on rational functions
- Cauchy principal value integrals

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula
- Gauss-Kronrod quadrature
- Gauss-Turán quadrature
- Quadrature formulae based on rational functions
- Cauchy principal value integrals
- Polynomials orthogonal on several intervals

Quadrature

- Gauss-type quadrature formulae
  - Gauss formula
  - Gauss-Radau formula
  - Gauss-Lobatto formula
- Gauss-Kronrod quadrature
- Gauss-Turán quadrature
- Quadrature formulae based on rational functions
- Cauchy principal value integrals
- Polynomials orthogonal on several intervals
- Quadrature estimates of matrix functionals

Approximation

Approximation

- Polynomial least squares approximation

Approximation

- Polynomial least squares approximation
  - classical

Approximation

- Polynomial least squares approximation
  - classical
  - constrained

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces
- Moment-preserving spline approximation

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces
- Moment-preserving spline approximation
  - on the positive real line

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces
- Moment-preserving spline approximation
  - on the positive real line
  - on a compact interval

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces
- Moment-preserving spline approximation
  - on the positive real line
  - on a compact interval
- Slowly convergent series

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces
- Moment-preserving spline approximation
  - on the positive real line
  - on a compact interval
- Slowly convergent series
  - generated by a Laplace transform or derivative thereof

Approximation

- Polynomial least squares approximation
  - classical
  - constrained
  - in Sobolev spaces
- Moment-preserving spline approximation
  - on the positive real line
  - on a compact interval
- Slowly convergent series
  - generated by a Laplace transform or derivative thereof
  - occurring in plate contact problems

References

References

- W. Gautschi, "Orthogonal Polynomials: Computation and Approximation", Oxford University Press, Oxford, 2004
- `http;//www.cs.purdue.edu/homes/wxg/papers.html/Madrid.ps` click on Madrid.ps

References

- W. Gautschi, "Orthogonal Polynomials: Computation and Approximation", Oxford University Press, Oxford, 2004
- `http;//www.cs.purdue.edu/homes/wxg/papers.html/Madrid.ps`
  click on Madrid.ps
- `http://www.cs.purdue.edu/archives/2002/wxg/codes`

Part I

Orthogonal Polynomials

Background

measure (of integration)

absolutely continuous measure

$$\mathrm{d}\lambda(t) = w(t)\mathrm{d}t \text{ on } [a, b], \quad -\infty \le a < b \le \infty$$

discrete measure

$$\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - x_k)\mathrm{d}t, \quad x_1 < x_2 < \cdots < x_N$$

Background

measure (of integration)

absolutely continuous measure

$$\mathrm{d}\lambda(t) = w(t)\mathrm{d}t \text{ on } [a, b], \quad -\infty \le a < b \le \infty$$

discrete measure

$$\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - x_k)\mathrm{d}t, \quad x_1 < x_2 < \cdots < x_N$$

standing assumption (for absolutely continuous measures)

existence of the moments

$$\mu_r = \int_{\mathbb{R}} t^r \mathrm{d}\lambda(t), \quad r = 0, 1, 2, \ldots$$

inner product and norm

$$(p,q)_{d\lambda} = \int_{\mathbb{R}} p(t)q(t)\,d\lambda(t), \quad (p,p)_{d\lambda} = \|p\|^2_{d\lambda}$$

inner product and norm

$$(p,q)_{d\lambda} = \int_{\mathbb{R}} p(t)q(t)d\lambda(t), \quad (p,p)_{d\lambda} = \|p\|^2_{d\lambda}$$

orthogonal polynomials

$$\pi_k(\,\cdot\,) = \pi_k(\,\cdot\,; d\lambda) \in \mathbb{P}_k,\ k = 0,1,2,\ldots$$

$$(\pi_k, \pi_\ell)_{d\lambda} \begin{cases} = 0, & k \neq \ell \\ > 0, & k = \ell \end{cases}$$

inner product and norm

$$(p,q)_{\mathrm{d}\lambda} = \int_{\mathbb{R}} p(t)q(t)\mathrm{d}\lambda(t), \quad (p,p)_{\mathrm{d}\lambda} = \|p\|^2_{\mathrm{d}\lambda}$$

orthogonal polynomials

$$\pi_k(\,\cdot\,) = \pi_k(\,\cdot\,; \mathrm{d}\lambda) \in \mathbb{P}_k,\ k = 0, 1, 2, \ldots$$

$$(\pi_k, \pi_\ell)_{\mathrm{d}\lambda} \begin{cases} = 0, & k \neq \ell \\ > 0, & k = \ell \end{cases}$$

orthonormal polynomials

$$\tilde{\pi}_k(\,\cdot\,; \mathrm{d}\lambda) = \frac{\pi_k(\,\cdot\,; \mathrm{d}\lambda)}{\|\pi_k\|_{\mathrm{d}\lambda}}, \quad k = 0, 1, 2, \ldots$$

three-term recurrence relation (for monic orth. pol.'s)

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t),\ k = 0, 1, \ldots, n-1$$

$$\alpha_k = \alpha_k(\mathrm{d}\lambda) \in \mathbb{R},\ \beta_k = \beta_k(\mathrm{d}\lambda) > 0 \quad \left(\beta_0 = \int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)$$

three-term recurrence relation (for monic orth. pol.'s)

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t),\ k = 0, 1, \dots, n-1$$
$$\alpha_k = \alpha_k(\mathrm{d}\lambda) \in \mathbb{R},\ \beta_k = \beta_k(\mathrm{d}\lambda) > 0 \quad \left(\beta_0 = \int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)$$

Proof Expand $\pi_{k+1}(t) - t\pi_k(t) \in \mathbb{P}_k$ in orthogonal polynomials $\pi_0, \pi_1, \dots, \pi_k$ and use orthogonality and the property $(tp, q)_{\mathrm{d}\lambda} = (p, tq)_{\mathrm{d}\lambda}$ of the inner product. The result is (Darboux)

$$\alpha_k(\mathrm{d}\lambda) = \frac{(t\pi_k, \pi_k)_{\mathrm{d}\lambda}}{(\pi_k, \pi_k)_{\mathrm{d}\lambda}}, \quad k = 0, 1, 2, \dots,$$
$$\beta_k(\mathrm{d}\lambda) = \frac{(\pi_k, \pi_k)_{\mathrm{d}\lambda}}{(\pi_{k-1}, \pi_{k-1})_{\mathrm{d}\lambda}}, \quad k = 1, 2, \dots \qquad \square$$

three-term recurrence relation (for monic orth. pol.'s)

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t),\ k = 0, 1, \ldots, n-1$$
$$\alpha_k = \alpha_k(\mathrm{d}\lambda) \in \mathbb{R},\ \beta_k = \beta_k(\mathrm{d}\lambda) > 0 \quad \left(\beta_0 = \int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)$$

Proof Expand $\pi_{k+1}(t) - t\pi_k(t) \in \mathbb{P}_k$ in orthogonal polynomials $\pi_0, \pi_1, \ldots, \pi_k$ and use orthogonality and the property $(tp, q)_{\mathrm{d}\lambda} = (p, tq)_{\mathrm{d}\lambda}$ of the inner product. The result is (Darboux)

$$\alpha_k(\mathrm{d}\lambda) = \frac{(t\pi_k, \pi_k)_{\mathrm{d}\lambda}}{(\pi_k, \pi_k)_{\mathrm{d}\lambda}}, \quad k = 0, 1, 2, \ldots,$$
$$\beta_k(\mathrm{d}\lambda) = \frac{(\pi_k, \pi_k)_{\mathrm{d}\lambda}}{(\pi_{k-1}, \pi_{k-1})_{\mathrm{d}\lambda}}, \quad k = 1, 2, \ldots \quad \square$$

normalization constants

$$\|\pi_k\|^2_{\mathrm{d}\lambda} = \beta_0 \beta_1 \cdots \beta_k$$

Jacobi matrix

of infinite order

$$\boldsymbol{J}(\mathrm{d}\lambda) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & \mathbf{0} \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \\ \mathbf{0} & & & & \end{bmatrix}$$

of order n

$$\boldsymbol{J}_n(\mathrm{d}\lambda) = \boldsymbol{J}(\mathrm{d}\lambda)_{[1:n,1:n]}$$

Jacobi matrix

of infinite order

$$\boldsymbol{J}(\mathrm{d}\lambda) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & \mathbf{0} \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \\ \mathbf{0} & & & & \end{bmatrix}$$

of order n

$$\boldsymbol{J}_n(\mathrm{d}\lambda) = \boldsymbol{J}(\mathrm{d}\lambda)_{[1:n,1:n]}$$

three-term recurrence relation (for orthonormal pol's)

in matrix form ($\tilde{\boldsymbol{\pi}}(t) = [\tilde{\pi}_0(t), \tilde{\pi}_1(t), \ldots, \tilde{\pi}_{n-1}(t)]^T$)

$$t\tilde{\boldsymbol{\pi}}(t) = \boldsymbol{J}_n(\mathrm{d}\lambda)\tilde{\boldsymbol{\pi}}(t) + \sqrt{\beta_n}\tilde{\pi}_n(t)\boldsymbol{e}_n$$

Jacobi matrix

of infi nite order

$$\boldsymbol{J}(\mathrm{d}\lambda) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & \boldsymbol{0} \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots \\ & & \ddots & \ddots \\ \boldsymbol{0} & & & \end{bmatrix}$$

of order n

$$\boldsymbol{J}_n(\mathrm{d}\lambda) = \boldsymbol{J}(\mathrm{d}\lambda)_{[1:n,1:n]}$$

three-term recurrence relation (for orthonormal pol's)

in matrix form ($\tilde{\boldsymbol{\pi}}(t) = [\tilde{\pi}_0(t), \tilde{\pi}_1(t), \ldots, \tilde{\pi}_{n-1}(t)]^T$)

$$t\tilde{\boldsymbol{\pi}}(t) = \boldsymbol{J}_n(\mathrm{d}\lambda)\tilde{\boldsymbol{\pi}}(t) + \sqrt{\beta_n}\tilde{\pi}_n(t)\boldsymbol{e}_n$$

$\Longrightarrow$ the zeros τ_ν of $\tilde{\pi}_n$ are the eigenvalues of $\boldsymbol{J}_n(\mathrm{d}\lambda)$ and $\tilde{\boldsymbol{\pi}}(\tau_\nu)$ corresponding eigenvectors

"Classical" weight functions

$$\mathrm{d}\lambda(t) = w(t)\mathrm{d}t$$

name	$w(t)$	support	comment
Jacobi	$(1-t)^\alpha(1+t)^\beta$	$[-1,1]$	$\alpha > -1,$ $\beta > -1$
Laguerre	$t^\alpha e^{-t}$	$[0,\infty]$	$\alpha > -1$
Hermite	$\|t\|^{2\alpha}e^{-t^2}$	$[-\infty,\infty]$	$\alpha > -\frac{1}{2}$
Meixner-Pollaczek	$\frac{1}{2\pi}\,e^{(2\phi-\pi)t}\|\Gamma(\lambda+it)\|^2$	$[-\infty,\infty]$	$\lambda > 0,$ $0 < \phi < \pi$

"Classical" discrete measures

$$\mathrm{d}\lambda(t) = \sum_{k=0}^{M} w_k \delta(t-k)\mathrm{d}t$$

name	M	w_k	comment
discrete Chebyshev	$N-1$	1	
Krawtchouk	N	$\binom{N}{k} p^k (1-p)^{N-k}$	$0 < p < 1$
Charlier	∞	$e^{-a} a^k / k!$	a>0
Meixner	∞	$\frac{c^k}{\Gamma(\beta)} \frac{\Gamma(k+\beta)}{k!}$	$0 < c < 1, \beta > 0$
Hahn	N	$\binom{\alpha+k}{k} \binom{\beta+N-k}{N-k}$	$\alpha > -1, \beta > -1$

Matlab

Example `ab=r_jacobi(N,a,b)`

$$\left\|\begin{array}{c|c} \alpha_0 & \beta_0 \\ \alpha_1 & \beta_1 \\ \vdots & \vdots \\ \alpha_{N-1} & \beta_{N-1} \end{array}\right\|$$

ab

$$N \in \mathbb{N},\ a > -1,\ b > -1$$

Demo #1

```
N=10;
ab=r_jacobi(N,-.5,1.5);
```

Demo #1

```
N=10;
ab=r_jacobi(N,-.5,1.5);
```

k	α_k	β_k
0	0.66666666666667	4.71238898038469
1	0.13333333333333	0.13888888888889
2	0.05714285714286	0.21000000000000
3	0.03174603174603	0.22959183673469
4	0.02020202020202	0.23765432098765
5	0.01398601398601	0.24173553719008
6	0.01025641025641	0.24408284023669
7	0.00784313725490	0.24555555555556
8	0.00619195046440	0.24653979238754
9	0.00501253132832	0.24722991689751

Modified Chebyshev algorithm

Modified Chebyshev algorithm

modified and mixed moments

$$m_k = \int_{\mathbb{R}} p_k(t) \mathrm{d}\lambda(t), \quad k = 0, 1, 2, \ldots$$

$$\sigma_{kl} = \int_{\mathbb{R}} \pi_k(t; \mathrm{d}\lambda) p_\ell(t) \mathrm{d}\lambda(t), \quad k, l \geq -1$$

Modified Chebyshev algorithm

modified and mixed moments

$$m_k = \int_{\mathbb{R}} p_k(t)\,\mathrm{d}\lambda(t), \quad k = 0, 1, 2, \ldots$$

$$\sigma_{kl} = \int_{\mathbb{R}} \pi_k(t; \mathrm{d}\lambda) p_\ell(t)\,\mathrm{d}\lambda(t), \quad k, l \geq -1$$

modified moment map

$$\mathbb{R}^{2n} \mapsto \mathbb{R}^{2n} : \quad [m_k]_{k=0}^{2n-1} \mapsto [\alpha_k, \beta_k]_{k=0}^{n-1}$$

Modified Chebyshev algorithm

modified and mixed moments

$$m_k = \int_{\mathbb{R}} p_k(t)\,\mathrm{d}\lambda(t), \quad k = 0, 1, 2, \ldots$$
$$\sigma_{kl} = \int_{\mathbb{R}} \pi_k(t; \mathrm{d}\lambda) p_\ell(t)\,\mathrm{d}\lambda(t), \quad k, l \geq -1$$

modified moment map

$$\mathbb{R}^{2n} \mapsto \mathbb{R}^{2n} : \quad [m_k]_{k=0}^{2n-1} \mapsto [\alpha_k, \beta_k]_{k=0}^{n-1}$$

conditioning: G., 1968, 1982, 2004
algorithm: Sack et al., 1971; Wheeler, 1974

assumption

$$p_{k+1}(t) = (t - a_k)p_k(t) - b_k p_{k-1}(t), \quad k = 0, 1, 2, \ldots$$

Modified Chebyshev algorithm (cont')

initialization:

$$\alpha_0 = a_0 + m_1/m_0, \quad \beta_0 = m_0,$$
$$\sigma_{-1,\ell} = 0, \quad \ell = 1, 2, \ldots, 2n-2,$$
$$\sigma_{0,\ell} = m_\ell, \quad \ell = 0, 1, \ldots, 2n-1$$

Modified Chebyshev algorithm (cont')

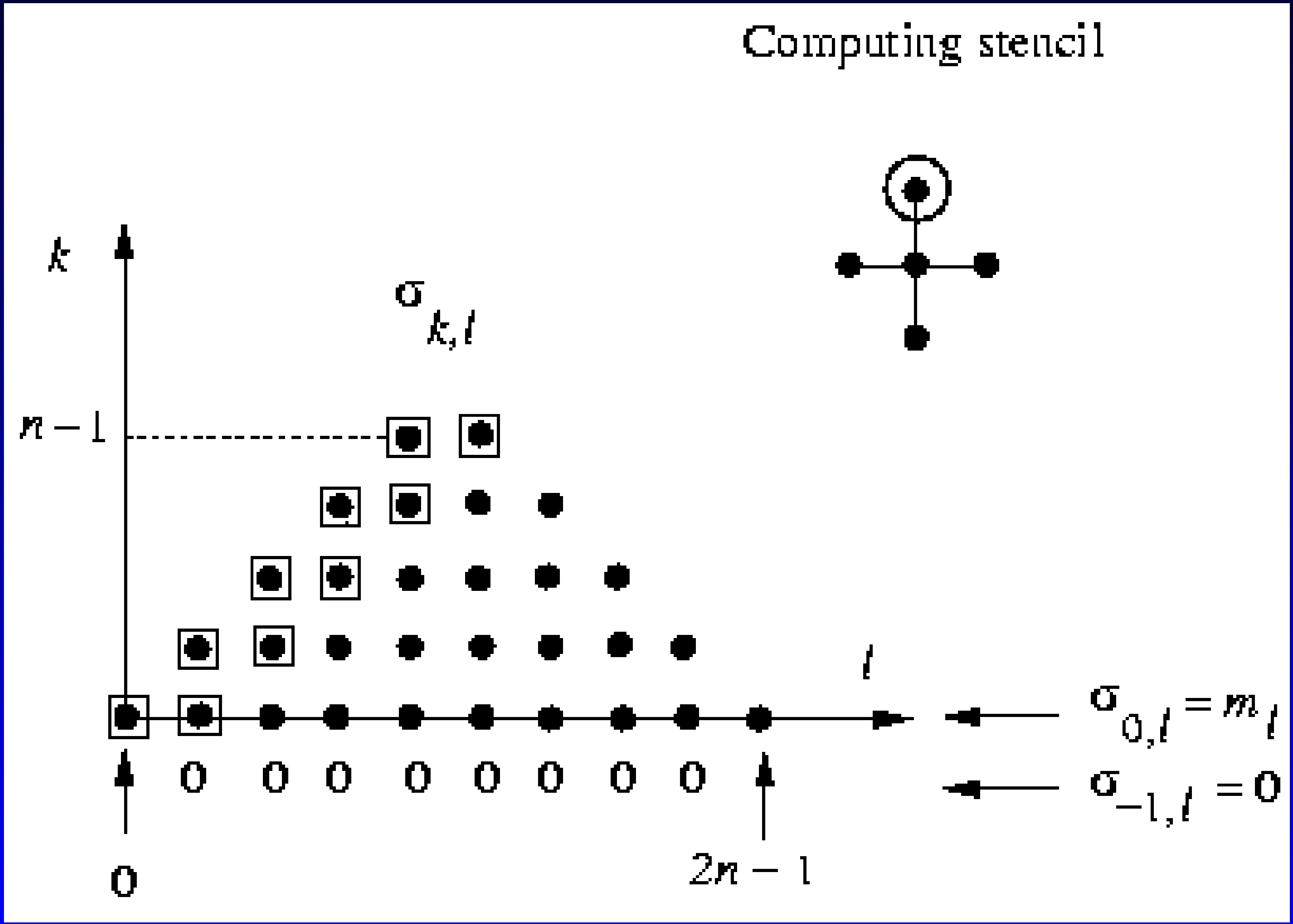
initialization:

$$\alpha_0 = a_0 + m_1/m_0, \quad \beta_0 = m_0,$$
$$\sigma_{-1,\ell} = 0, \quad \ell = 1, 2, \ldots, 2n-2,$$
$$\sigma_{0,\ell} = m_\ell, \quad \ell = 0, 1, \ldots, 2n-1$$

continuation (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$\sigma_{k\ell} = \sigma_{k-1,\ell+1} - (\alpha_{k-1} - a_\ell)\sigma_{k-1,\ell} - \beta_{k-1}\sigma_{k-2,\ell}$$
$$+ b_\ell \sigma_{k-1,\ell-1}, \quad \ell = k, k+1, \ldots, 2n-k-1,$$
$$\alpha_k = a_k + \frac{\sigma_{k,k+1}}{\sigma_{kk}} - \frac{\sigma_{k-1,k}}{\sigma_{k-1,\ell-1}}, \quad \beta_k = \frac{\sigma_{kk}}{\sigma_{k-1,k-1}}$$

Modified Chebyshev algorithm, schematically



Matlab

m_0	m_1	m_2	$\cdots$	m_{2N-1}

`mom`

a_0	b_0
a_1	b_1
$\vdots$	$\vdots$
a_{2N-2}	b_{2N-2}

`abm`

```
ab=chebyshev(N,mom,abm)
```

Demo #2

elliptic orthogonal polynomials

$$\mathrm{d}\lambda(t) = [(1-\omega^2t^2)(1-t^2)]^{-1/2}\mathrm{d}t \ \text{ on } [-1,1],$$
$$0 \le \omega < 1$$

Demo #2

elliptic orthogonal polynomials

$$\mathrm{d}\lambda(t) = [(1-\omega^2 t^2)(1-t^2)]^{-1/2}\mathrm{d}t \;\; \text{on } [-1,1], \quad 0 \le \omega < 1$$

Chebyshev moments

$$m_0 = \int_{-1}^{1} \mathrm{d}\lambda(t), \;\; m_k = \frac{1}{2^{k-1}} \int_{-1}^{1} T_k(t)\mathrm{d}\lambda(t), \;\; k \ge 1$$

Demo #2

elliptic orthogonal polynomials

$$\mathrm{d}\lambda(t) = [(1-\omega^2 t^2)(1-t^2)]^{-1/2}\mathrm{d}t \ \text{ on } [-1,1],$$
$$0 \le \omega < 1$$

Chebyshev moments

$$m_0 = \int_{-1}^{1} \mathrm{d}\lambda(t), \ \ m_k = \frac{1}{2^{k-1}} \int_{-1}^{1} T_k(t)\mathrm{d}\lambda(t), \ \ k \ge 1$$

Matlab

```
function ab=r_elliptic(N,om2)
   abm=r_jacobi(2*N-1,-1/2);
   mom=mm_elliptic(N,om2);
   ab=chebyshev(N,mom,abm);
```

Demo #2 (cont')

$$\omega^2 = .999, \quad N = 40$$

Demo #2 (cont’)

$$\omega^2 = .999, \quad N = 40$$

k	β_k	k	β_k	k	β_k	k	β_k
0	9.68226512	10	0.24936494	20	0.24992062	30	0.24998062
1	0.79378214	11	0.24951641	21	0.24993230	31	0.24998288
2	0.11986767	12	0.24962381	22	0.24994197	32	0.24998485
3	0.22704012	13	0.24970218	23	0.24995003	33	0.24998657
4	0.24106088	14	0.24976074	24	0.24995679	34	0.24998806
5	0.24542853	15	0.24980537	25	0.24996249	35	0.24998937
6	0.24730165	16	0.24983998	26	0.24996732	36	0.24999052
7	0.24825871	17	0.24986721	27	0.24997145	37	0.24999154
8	0.24880566	18	0.24988890	28	0.24997497	38	0.24999243
9	0.24914365	19	0.24990639	29	0.24997800	39	0.24999322

Discrete measure

$$\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - x_k)\mathrm{d}t$$

inner product $\quad (p, q)_N = \sum_{k=1}^{N} w_k p(x_k) q(x_k)$

Discrete measure

$$\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - x_k)\mathrm{d}t$$

inner product $\quad (p, q)_N = \sum_{k=1}^{N} w_k p(x_k) q(x_k)$

Algorithm #1: Stieltjes, 1884; G., 1968

Darboux's formulae

$$(D) \qquad \begin{cases} \alpha_k = \frac{(t\pi_k, \pi_k)_N}{(\pi_k, \pi_k)_N}, & k = 0, 1, \ldots, n-1, \\ \beta_k = \frac{(\pi_k, \pi_k)_N}{(\pi_{k-1}, \pi_{k-1})_N}, & k = 1, 2, \ldots, n-1 \end{cases}$$

Discrete measure

$$d\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - x_k)\mathrm{d}t$$

inner product $\quad (p, q)_N = \sum_{k=1}^{N} w_k p(x_k) q(x_k)$

Algorithm #1: Stieltjes, 1884; G., 1968

Darboux's formulae

$$(D) \qquad \begin{cases} \alpha_k = \frac{(t\pi_k, \pi_k)_N}{(\pi_k, \pi_k)_N}, & k = 0, 1, \ldots, n-1, \\ \beta_k = \frac{(\pi_k, \pi_k)_N}{(\pi_{k-1}, \pi_{k-1})_N}, & k = 1, 2, \ldots, n-1 \end{cases}$$

plus recurrence relation (R)

$$\pi_0 = 1 \overset{(D)}{\Longrightarrow} \alpha_0, \beta_0 \overset{(R)}{\Longrightarrow} \pi_1 \overset{(D)}{\Longrightarrow} \alpha_1, \beta_1 \overset{(R)}{\Longrightarrow} \cdots \overset{(D)}{\Longrightarrow} \alpha_{n-1}, \beta_{n-1}$$

Algorithm #2 Lanczos, 1950

$$\boldsymbol{Q}^T \begin{bmatrix} 1 & \sqrt{w_1} & \sqrt{w_2} & \cdots & \sqrt{w_N} \\ \sqrt{w_1} & x_1 & 0 & \cdots & 0 \\ \sqrt{w_2} & 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sqrt{w_N} & 0 & 0 & \cdots & x_N \end{bmatrix} \boldsymbol{Q}$$

$$= \begin{bmatrix} 1 & \sqrt{\beta_0} & 0 & \cdots & 0 \\ \sqrt{\beta_0} & \alpha_0 & \sqrt{\beta_1} & \cdots & 0 \\ 0 & \sqrt{\beta_1} & \alpha_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_{N-1} \end{bmatrix}$$

stable version of Lanczos algorithm

Rutishauser, 1963; Gragg and Harrod, 1984

Matlab

```
ab=stieltjes(n,xw)
ab=lanczos(n,xw)
```

$\left.\right\}\quad n \leq N$

x_1	w_1
x_2	w_2
$\vdots$	$\vdots$
x_N	w_N

`xw`

Discretization methods G.; 1968, 1982

Discretization methods **G.; 1968, 1982**

basic idea

$$\mathrm{d}\lambda(t) \approx \mathrm{d}\lambda_N(t), \qquad \begin{cases} \alpha_k(\mathrm{d}\lambda) \approx \alpha_k(\mathrm{d}\lambda_N) \\ \beta_k(\mathrm{d}\lambda) \approx \beta_k(\mathrm{d}\lambda_N) \end{cases}$$

Discretization methods **G.; 1968, 1982**

basic idea

$$\mathrm{d}\lambda(t) \approx \mathrm{d}\lambda_N(t), \quad \begin{cases} \alpha_k(\mathrm{d}\lambda) \approx \alpha_k(\mathrm{d}\lambda_N) \\ \beta_k(\mathrm{d}\lambda) \approx \beta_k(\mathrm{d}\lambda_N) \end{cases}$$

Example

$$w(t) = (1-t^2)^{-1/2} + c \ \text{ on } [-1,1], \quad c>0$$

Discretization methods **G.; 1968, 1982**

basic idea

$$\mathrm{d}\lambda(t) \approx \mathrm{d}\lambda_N(t), \quad \begin{cases} \alpha_k(\mathrm{d}\lambda) \approx \alpha_k(\mathrm{d}\lambda_N) \\ \beta_k(\mathrm{d}\lambda) \approx \beta_k(\mathrm{d}\lambda_N) \end{cases}$$

Example

$$w(t) = (1-t^2)^{-1/2} + c \ \text{ on } [-1,1], \quad c > 0$$

discretization

$$(p,q) =$$

$$\int_{-1}^{1} p(t)q(t)(1-t^2)^{-1/2}\mathrm{d}t + c\int_{-1}^{1} p(t)q(t)\mathrm{d}t$$

$$\approx \sum_{k=1}^{N} w_k^{Ch} p(x_k^{Ch}) q(x_k^{Ch}) + c\sum_{k=1}^{N} w_k^{L} p(x_k^{L}) q(x_k^{L})$$

in general

discretize $\mathrm{d}\lambda$ on

$$\bigcup_{j=1}^{s} [a_j, b_j]$$

using

$$\left.\begin{array}{l}\text{tailor-made quadratures} \\[1em] \text{general-purpose quadratures}\end{array}\right\} \text{ on each } [a_j, b_j]$$

Matlab

`ab=mcdis(n,eps0,quad,Nmax)`

global variables to describe details of discretization

$$mc,\ mp,\ iq$$

$$AB = \begin{array}{||c|c||}\hline\hline a_1 & b_1 \\ a_2 & b_2 \\ \vdots & \vdots \\ a_{\mathrm{mc}} & b_{\mathrm{mc}} \\ \hline\hline\end{array},\ DM = \begin{array}{||c|c||}\hline\hline x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{\mathrm{mp}} & y_{\mathrm{mp}} \\ \hline\hline\end{array}$$

Example

Jacobi weight function plus a discrete measure

$$\mathrm{d}\lambda(t) = (\beta_0^J)^{-1}(1-t)^\alpha(1+t)^\beta \mathrm{d}t + \sum_{j=1}^{p} w_j \delta(t-x_j)\mathrm{d}t \text{ on } [-1,1]$$

$$\beta_0^J = \int_{-1}^{1}(1-t)^\alpha(1+t)^\beta \mathrm{d}t, \quad \alpha > -1,\ \beta > -1$$

Example

Jacobi weight function plus a discrete measure

$$\mathrm{d}\lambda(t) = (\beta_0^J)^{-1}(1-t)^\alpha(1+t)^\beta \mathrm{d}t + \sum_{j=1}^{p} w_j \delta(t-x_j)\mathrm{d}t \text{ on } [-1,1]$$

$$\beta_0^J = \int_{-1}^{1} (1-t)^\alpha(1+t)^\beta \mathrm{d}t, \quad \alpha > -1,\ \beta > -1$$

discretization (yields $\alpha_k(\mathrm{d}\lambda)$, $\beta_k(\mathrm{d}\lambda)$ exactly for $k \le n-1$)

$$\int_{-1}^{1} p(t)\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n} \lambda_\nu^J p(\tau_\nu^J) + \sum_{j=1}^{p} w_j p(x_j)$$

Example

Jacobi weight function plus a discrete measure

$$d\lambda(t) = (\beta_0^J)^{-1}(1-t)^\alpha(1+t)^\beta dt + \sum_{j=1}^{p} w_j \delta(t-x_j)dt \text{ on } [-1,1]$$

$$\beta_0^J = \int_{-1}^{1}(1-t)^\alpha(1+t)^\beta dt, \quad \alpha>-1,\ \beta>-1$$

discretization (yields $\alpha_k(d\lambda)$, $\beta_k(d\lambda)$ exactly for $k \le n-1$)

$$\int_{-1}^{1} p(t)d\lambda(t) \approx \sum_{\nu=1}^{n} \lambda_\nu^J p(\tau_\nu^J) + \sum_{j=1}^{p} w_j p(x_j)$$

here, `mc`=1, `mp`=p, `iq`=1 (if $x_j \in [-1,1]$), and

AB= $\begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array}$ DM= $\begin{array}{|c|c|} \hline x_1 & w_1 \\ x_2 & w_2 \\ \vdots & \vdots \\ x_p & w_p \\ \hline \end{array}$

Demo #3 **logistic density function**

$$\mathrm{d}\lambda(t) = \frac{e^{-t}}{(1+e^{-t})^2}, \quad t \in \mathbb{R}$$

Demo #3 **logistic density function**

$$\mathrm{d}\lambda(t) = \frac{e^{-t}}{(1+e^{-t})^2}, \quad t \in \mathbb{R}$$

discretization

$$\int_{\mathbb{R}} p(t)\mathrm{d}\lambda(t) = \int_0^\infty \frac{p(-t)}{(1+e^{-t})^2}\, e^{-t}\mathrm{d}t + \int_0^\infty \frac{p(t)}{(1+e^{-t})^2}\, e^{-t}\mathrm{d}t$$

$$\approx \sum_{\nu=1}^{n} \lambda_\nu^L \frac{p(-\tau_\nu^L)+p(\tau_\nu^L)}{(1+e^{-\tau_\nu^L})^2}$$

Demo #3 logistic density function

$$\mathrm{d}\lambda(t) = \frac{e^{-t}}{(1+e^{-t})^2}, \quad t \in \mathbb{R}$$

discretization

$$\int_{\mathbb{R}} p(t)\mathrm{d}\lambda(t) = \int_0^\infty \frac{p(-t)}{(1+e^{-t})^2}\, e^{-t}\mathrm{d}t + \int_0^\infty \frac{p(t)}{(1+e^{-t})^2}\, e^{-t}\mathrm{d}t$$
$$\approx \sum_{\nu=1}^{n} \lambda_\nu^L \frac{p(-\tau_\nu^L)+p(\tau_\nu^L)}{(1+e^{-\tau_\nu^L})^2}$$

exact answers

$\alpha_k(\mathrm{d}\lambda) = 0, \quad \beta_0 = 1, \quad \beta_k = k^4\pi^2/(4k^2-1), \quad k \geq 1$

numerical results (`N=40`, `eps0=1000` $\times$`eps`)

n	β_n	err α	err β
0	1.0000000000(0)	7.18(−17)	3.33(−16)
1	3.2898681337(0)	1.29(−16)	2.70(−16)
6	8.9117603523(1)	4.52(−16)	1.43(−15)
15	5.5578278399(2)	2.14(−14)	0.00(+00)
39	3.7535340252(3)	6.24(−14)	4.48(−15)
		6.24(−14)	8.75(−15)

Cauchy integrals of orthogonal polynomials

$$\rho_n(z) = \rho_n(z; \mathrm{d}\lambda) = \int_a^b \frac{\pi_n(t; \mathrm{d}\lambda)}{z - t}\, \mathrm{d}\lambda(t), \quad z \in \mathbb{C}\backslash[a, b],$$

where $[a, b] = \mathrm{supp}(\mathrm{d}\lambda)$. By convention $\rho_{-1}(z; \mathrm{d}\lambda) = 1$.

Cauchy integrals of orthogonal polynomials

$$\rho_n(z) = \rho_n(z; \mathrm{d}\lambda) = \int_a^b \frac{\pi_n(t; \mathrm{d}\lambda)}{z - t} \, \mathrm{d}\lambda(t), \quad z \in \mathbb{C} \backslash [a, b],$$

where $[a, b] = \mathrm{supp}(\mathrm{d}\lambda)$. By convention $\rho_{-1}(z; \mathrm{d}\lambda) = 1$.

Let $\alpha_k = \alpha_k(\mathrm{d}\lambda)$, $\beta_k = \beta_k(\mathrm{d}\lambda)$, $k = 0, 1, 2, \dots$.

Theorem (G., 1981). If $z \in \mathbb{C}\backslash[a, b]$, then $\{\rho_k(z)\}_{k=-1}^{\infty}$ is the minimal solution of the three-term recurrence relation

$$y_{k+1} = (z - \alpha_k) y_k - \beta_k y_{k-1}, \quad k = 0, 1, 2, \dots ,$$

with initial value $y_{-1} = 1$. $\square$

Cauchy integrals of orthogonal polynomials

$$\rho_n(z) = \rho_n(z; d\lambda) = \int_a^b \frac{\pi_n(t; d\lambda)}{z-t}\, d\lambda(t), \quad z \in \mathbb{C}\backslash[a,b],$$

where $[a,b] = \mathrm{supp}(d\lambda)$. By convention $\rho_{-1}(z; d\lambda) = 1$.

Let $\alpha_k = \alpha_k(d\lambda)$, $\beta_k = \beta_k(d\lambda)$, $k = 0, 1, 2, \ldots$.

Theorem (G., 1981). If $z \in \mathbb{C}\backslash[a,b]$, then $\{\rho_k(z)\}_{k=-1}^{\infty}$ is the minimal solution of the three-term recurrence relation

$$y_{k+1} = (z - \alpha_k)y_k - \beta_k y_{k-1}, \quad k = 0, 1, 2, \ldots ,$$

with initial value $y_{-1} = 1$. □

If $z = x \in (a,b)$, the integrals are Cauchy principal value integrals. They satisfy the same three-term recurrence relation, but with initial values $\rho_{-1}(x) = 1$, $\rho_0(x) = ⨍_a^b d\lambda(t)/(x-t)$.

Continued fraction algorithm (G., 1967)

Formally,

$$r_{n-1} := \frac{\rho_n(z)}{\rho_{n-1}(z)} = \frac{\beta_n}{z-\alpha_n-} \; \frac{\beta_{n+1}}{z-\alpha_{n+1}-} \; \frac{\beta_{n+2}}{z-\alpha_{n+2}-} \; \cdots$$

Continued fraction algorithm (G., 1967)

Formally,

$$r_{n-1} := \frac{\rho_n(z)}{\rho_{n-1}(z)} = \frac{\beta_n}{z-\alpha_n-}\;\frac{\beta_{n+1}}{z-\alpha_{n+1}-}\;\frac{\beta_{n+2}}{z-\alpha_{n+2}-}\;\cdots$$

Theorem (Pincherle, 1894). The continued fraction converges if and only if $\{\rho_k(z)\}_{k=-1}^{\infty}$ is a minimal solution of the three-term recurrence relation. $\square$

Continued fraction algorithm (G., 1967)

Formally,

$$r_{n-1} := \frac{\rho_n(z)}{\rho_{n-1}(z)} = \frac{\beta_n}{z-\alpha_n-}\,\frac{\beta_{n+1}}{z-\alpha_{n+1}-}\,\frac{\beta_{n+2}}{z-\alpha_{n+2}-}\cdots$$

Theorem (Pincherle, 1894). The continued fraction converges if and only if $\{\rho_k(z)\}_{k=-1}^{\infty}$ is a minimal solution of the three-term recurrence relation. □

To compute $\rho_n(z)$, $n=-1,0,\ldots,N$, suppose first that r_ν is known for some $\nu \geq N$. Then

$$r_{n-1} = \frac{\beta_n}{z-\alpha_n-r_n}, \quad n=\nu,\nu-1,\ldots,0,$$

and

$$\rho_{-1}(z)=1,\ \ \rho_n(z)=r_{n-1}\rho_{n-1}(z), \quad n=0,1,\ldots,N$$

Continued fraction algorithm (G., 1967)

Formally,

$$r_{n-1} := \frac{\rho_n(z)}{\rho_{n-1}(z)} = \frac{\beta_n}{z-\alpha_n-}\;\frac{\beta_{n+1}}{z-\alpha_{n+1}-}\;\frac{\beta_{n+2}}{z-\alpha_{n+2}-}\;\cdots$$

Theorem (Pincherle, 1894). The continued fraction converges if and only if $\{\rho_k(z)\}_{k=-1}^{\infty}$ is a minimal solution of the three-term recurrence relation. $\square$

To compute $\rho_n(z)$, $n=-1,0,\ldots,N$, suppose first that r_ν is known for some $\nu \geq N$. Then

$$r_{n-1} = \frac{\beta_n}{z-\alpha_n-r_n}, \quad n=\nu,\nu-1,\ldots,0,$$

and

$$\rho_{-1}(z)=1,\;\; \rho_n(z)=r_{n-1}\rho_{n-1}(z), \quad n=0,1,\ldots,N$$

If r_ν is not known, replace it by $r_\nu \approx 0$.

Algorithm

Defi ne

$$r_\nu^{[\nu]} = 0, \quad r_{n-1}^{[\nu]} = \frac{\beta_n}{z-\alpha_n-} \, \frac{\beta_{n+1}}{z-\alpha_{n+1}-} \, \cdots \, \frac{\beta_\nu}{z-\alpha_\nu}$$

Then

$$r_\nu^{[\nu]} = 0, \;\; r_{n-1}^{[\nu]} = \frac{\beta_n}{z - \alpha_n - r_n^{[\nu]}}, \quad n = \nu, \nu - 1 \ldots, 0,$$

$$\rho_{-1}^{[\nu]} = 1, \;\; \rho_n^{[\nu]} = r_{n-1}^{[\nu]} \rho_{n-1}^{[\nu]}, \quad n = 0, 1, \ldots, N$$

Algorithm

Define

$$r_\nu^{[\nu]} = 0, \quad r_{n-1}^{[\nu]} = \frac{\beta_n}{z-\alpha_n-} \frac{\beta_{n+1}}{z-\alpha_{n+1}-} \cdots \frac{\beta_\nu}{z-\alpha_\nu}$$

Then

$$r_\nu^{[\nu]} = 0, \;\; r_{n-1}^{[\nu]} = \frac{\beta_n}{z-\alpha_n-r_n^{[\nu]}}, \quad n = \nu, \nu-1 \ldots, 0,$$

$$\rho_{-1}^{[\nu]} = 1, \;\; \rho_n^{[\nu]} = r_{n-1}^{[\nu]} \rho_{n-1}^{[\nu]}, \quad n = 0, 1, \ldots, N$$

Theorem (G., 1981) If $z \in \mathbb{C} \backslash [a, b]$, then

$$\lim_{\nu\to\infty} \rho_n^{[\nu]} = \rho_n(z), \quad n = 0, 1, \ldots, N. \; \square$$

Algorithm

Define

$$r_\nu^{[\nu]} = 0, \quad r_{n-1}^{[\nu]} = \frac{\beta_n}{z-\alpha_n-}\,\frac{\beta_{n+1}}{z-\alpha_{n+1}-}\cdots\frac{\beta_\nu}{z-\alpha_\nu}$$

Then

$$r_\nu^{[\nu]} = 0, \; r_{n-1}^{[\nu]} = \frac{\beta_n}{z-\alpha_n-r_n^{[\nu]}}, \quad n = \nu, \nu-1 \ldots, 0,$$

$$\rho_{-1}^{[\nu]} = 1, \; \rho_n^{[\nu]} = r_{n-1}^{[\nu]}\rho_{n-1}^{[\nu]}, \quad n = 0, 1, \ldots, N$$

Theorem (G., 1981) If $z \in \mathbb{C}\backslash[a, b]$, then

$$\lim_{\nu\to\infty} \rho_n^{[\nu]} = \rho_n(z), \quad n = 0, 1, \ldots, N. \;\square$$

Convergence is faster the larger dist$(z, [a, b])$.

Matlab

```
[rho,r,nu]=cauchy(N,ab,z,eps0,nu0,num
```

$$\left\|\begin{array}{c}\rho_0(z)\\ \rho_1(z)\\ \vdots\\ \rho_N(z)\end{array}\right\| \qquad \left\|\begin{array}{c}r_0(z)\\ r_1(z)\\ \vdots\\ r_N(z)\end{array}\right\| \qquad \left\|\begin{array}{c|c}\alpha_0 & \beta_0\\ \alpha_1 & \beta_1\\ \vdots & \vdots\\ \alpha_{\mathrm{numax}} & \beta_{\mathrm{numax}}\end{array}\right\|$$

`rho` `r` `ab`

where

$$r_k(z) = \frac{\rho_{k+1}(z)}{\rho_k(z)}, \quad k = 0, 1, \ldots, N$$

Modification algorithms

Modification algorithms

Problem: given the recurrence coefficients of $\mathrm{d}\lambda$, generate those of

$$\mathrm{d}\hat{\lambda}(t) = r(t)\mathrm{d}\lambda(t),\ r \geq 0 \text{ on supp}(\mathrm{d}\lambda),\ \text{rational}$$

Modification algorithms

Problem: given the recurrence coefficients of $\mathrm{d}\lambda$, generate those of

$$\mathrm{d}\hat{\lambda}(t) = r(t)\mathrm{d}\lambda(t),\ r \geq 0 \text{ on } \mathrm{supp}(\mathrm{d}\lambda),\ \text{rational}$$

Example: Galant, 1971

$$r(t) = s(t-c), \quad c \in \mathbb{R}\backslash\mathrm{supp}(\mathrm{d}\lambda), \quad s = \pm 1$$

Modification algorithms

Problem: given the recurrence coefficients of $\mathrm{d}\lambda$, generate those of

$$\mathrm{d}\hat{\lambda}(t) = r(t)\mathrm{d}\lambda(t),\ r \geq 0 \text{ on } \operatorname{supp}(\mathrm{d}\lambda),\ \text{rational}$$

Example: Galant, 1971

$$r(t) = s(t-c), \quad c \in \mathbb{R}\backslash\operatorname{supp}(\mathrm{d}\lambda), \quad s = \pm 1$$

one step of (symmetric, shifted) LR algorithm:

$$s[\boldsymbol{J}_{n+1}(\mathrm{d}\lambda) - c\boldsymbol{I}] = \boldsymbol{L}\boldsymbol{L}^T$$
$$\boldsymbol{J}_n(\mathrm{d}\hat{\lambda}) = \left(\boldsymbol{L}^T\boldsymbol{L} + c\boldsymbol{I}\right)_{[1:n,1:n]}$$

nonlinear recurrence algorithm

generalized Christoffel theorem (Uvarov, 1969)

$$\mathrm{d}\hat{\lambda}(t) = \frac{u(t)}{v(t)}\,\mathrm{d}t,\; u(t) = \pm\prod_{\lambda=1}^{\ell}(t-u_\lambda),\; v(t) = \prod_{\mu=1}^{m}(t-v_\mu)$$

generalized Christoffel theorem (Uvarov, 1969)

$$\mathrm{d}\hat{\lambda}(t) = \frac{u(t)}{v(t)}\,\mathrm{d}t,\ u(t) = \pm\prod_{\lambda=1}^{\ell}(t-u_\lambda),\ v(t) = \prod_{\mu=1}^{m}(t-v_\mu)$$

Let $m \le n$. Then

$$u(t)\pi_n(t;\mathrm{d}\hat{\lambda}) = \text{const} \times$$

$$\begin{vmatrix} \pi_{n-m}(t) & \cdots & \pi_{n-1}(t) & \pi_n(t) & \cdots & \pi_{n+\ell}(t) \\ \pi_{n-m}(u_1) & \cdots & \pi_{n-1}(u_1) & \pi_n(u_1) & \cdots & \pi_{n+\ell}(u_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi_{n-m}(u_\ell) & \cdots & \pi_{n-1}(u_\ell) & \pi_n(u_\ell) & \cdots & \pi_{n+\ell}(u_\ell) \\ \rho_{n-m}(v_1) & \cdots & \rho_{n-1}(v_1) & \rho_n(v_1) & \cdots & \rho_{n+\ell}(v_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{n-m}(v_m) & \cdots & \rho_{n-1}(v_m) & \rho_n(v_m) & \cdots & \rho_{n+\ell}(v_m) \end{vmatrix}$$

where

$$\rho_k(z) = \int_{\mathbb{R}} \frac{\pi_k(t;\mathrm{d}\lambda)}{z-t}\,\mathrm{d}\lambda(t), \quad k = 0, 1, 2, \ldots\ . \ \square$$

generalized Christoffel theorem **(Uvarov, 1969)**

$$\mathrm{d}\hat{\lambda}(t) = \frac{u(t)}{v(t)}\,\mathrm{d}t,\ u(t) = \pm\prod_{\lambda=1}^{\ell}(t-u_\lambda),\ v(t) = \prod_{\mu=1}^{m}(t-v_\mu)$$

Let $m \le n$. Then

$$u(t)\pi_n(t;\mathrm{d}\hat{\lambda}) = \text{const} \times \begin{vmatrix} \pi_{n-m}(t) & \cdots & \pi_{n-1}(t) & \pi_n(t) & \cdots & \pi_{n+\ell}(t) \\ \pi_{n-m}(u_1) & \cdots & \pi_{n-1}(u_1) & \pi_n(u_1) & \cdots & \pi_{n+\ell}(u_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi_{n-m}(u_\ell) & \cdots & \pi_{n-1}(u_\ell) & \pi_n(u_\ell) & \cdots & \pi_{n+\ell}(u_\ell) \\ \rho_{n-m}(v_1) & \cdots & \rho_{n-1}(v_1) & \rho_n(v_1) & \cdots & \rho_{n+\ell}(v_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{n-m}(v_m) & \cdots & \rho_{n-1}(v_m) & \rho_n(v_m) & \cdots & \rho_{n+\ell}(v_m) \end{vmatrix}$$

where

$$\rho_k(z) = \int_{\mathbb{R}} \frac{\pi_k(t;\mathrm{d}\lambda)}{z-t}\,\mathrm{d}\lambda(t), \quad k = 0, 1, 2, \ldots\ . \ \square$$

Christoffel's theorem: $m = 0$

Theorem (Uvarov, 1969) Let $m > n$. Then

$$u(t)\pi_n(t; d\hat{\lambda}) = \text{const} \times$$

$$\begin{vmatrix} 0 & 0 & \cdots & 0 & \pi_0(t) & \cdots & \pi_{n+\ell}(t) \\ 0 & 0 & \cdots & 0 & \pi_0(u_1) & \cdots & \pi_{n+\ell}(u_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \pi_0(u_\ell) & \cdots & \pi_{n+\ell}(u_\ell) \\ 1 & v_1 & \cdots & v_1^{m-n-1} & \rho_0(v_1) & \cdots & \rho_{n+\ell}(v_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & v_m & \cdots & v_m^{m-n-1} & \rho_0(v_m) & \cdots & \rho_{n+\ell}(v_m) \end{vmatrix}$$

where $\rho_k(z)$ is the Cauchy integral defined previously. $\square$

Theorem (Uvarov, 1969) Let $m > n$. Then

$$u(t)\pi_n(t; \mathrm{d}\hat{\lambda}) = \text{const} \times \begin{vmatrix} 0 & 0 & \cdots & 0 & \pi_0(t) & \cdots & \pi_{n+\ell}(t) \\ 0 & 0 & \cdots & 0 & \pi_0(u_1) & \cdots & \pi_{n+\ell}(u_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \pi_0(u_\ell) & \cdots & \pi_{n+\ell}(u_\ell) \\ 1 & v_1 & \cdots & v_1^{m-n-1} & \rho_0(v_1) & \cdots & \rho_{n+\ell}(v_1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & v_m & \cdots & v_m^{m-n-1} & \rho_0(v_m) & \cdots & \rho_{n+\ell}(v_m) \end{vmatrix}$$

where $\rho_k(z)$ is the Cauchy integral defi ned previously. $\square$

The generalized Christoffel theorems remain valid for complex u_λ, v_μ if orthogonality is understodd in the sense of formal orthogonality.

linear factor $\mathrm{d}\hat{\lambda}(t) = (t-z)\mathrm{d}\lambda(t),\ z \in \mathbb{C}\backslash[a,b]$

Christoffel:

$$(t-z)\hat{\pi}_n(t) = \frac{\begin{vmatrix} \pi_n(t) & \pi_{n+1}(t) \\ \pi_n(z) & \pi_{n+1}(z) \end{vmatrix}}{-\pi_n(z)} = \pi_{n+1}(t) - r_n\pi_n(t),\ r_n = \frac{\pi_{n+1}(z)}{\pi_n(z)}$$

linear factor $d\hat{\lambda}(t) = (t-z)d\lambda(t),\ z \in \mathbb{C}\backslash[a,b]$

Christoffel:

$$(t-z)\hat{\pi}_n(t) = \frac{\begin{vmatrix} \pi_n(t) & \pi_{n+1}(t) \\ \pi_n(z) & \pi_{n+1}(z) \end{vmatrix}}{-\pi_n(z)} = \pi_{n+1}(t) - r_n\pi_n(t),\ r_n = \frac{\pi_{n+1}(z)}{\pi_n(z)}$$

write $(t-z)t\hat{\pi}_k(t)$ in two ways (Verlinden, 1999)

$$(t-z)t\hat{\pi}_k(t) = t\pi_{k+1}(t) - r_k \cdot t\pi_k(t)$$

$$= \pi_{k+2}(t) + (\alpha_{k+1} - r_k)\pi_{k+1}(t) + (\beta_{k+1} - r_k\alpha_k)\pi_k(t) - r_k\beta_k\pi_{k-1}(t)$$

$$(t-z)t\hat{\pi}_k(t) = (t-z)[\hat{\pi}_{k+1} + \hat{\alpha}_k\hat{\pi}_k(t) + \hat{\beta}_k\hat{\pi}_{k-1}(t)]$$

$$= \pi_{k+2}(t) + (\hat{\alpha}_k - r_{k+1})\pi_{k+1}(t) + (\hat{\beta}_k - r_k\hat{\alpha}_k)\pi_k(t) - r_{k-1}\hat{\beta}_k\pi_{k-1}(t)$$

linear factor $\mathrm{d}\hat{\lambda}(t) = (t-z)\mathrm{d}\lambda(t),\ z \in \mathbb{C}\backslash[a,b]$

Christoffel:

$$(t-z)\hat{\pi}_n(t) = \frac{\begin{vmatrix} \pi_n(t) & \pi_{n+1}(t) \\ \pi_n(z) & \pi_{n+1}(z) \end{vmatrix}}{-\pi_n(z)} = \pi_{n+1}(t) - r_n\pi_n(t),\ r_n = \frac{\pi_{n+1}(z)}{\pi_n(z)}$$

write $(t-z)t\hat{\pi}_k(t)$ in two ways (Verlinden, 1999)

$$(t-z)t\hat{\pi}_k(t) = t\pi_{k+1}(t) - r_k \cdot t\pi_k(t)$$

$$= \pi_{k+2}(t) + (\alpha_{k+1} - r_k)\pi_{k+1}(t) + (\beta_{k+1} - r_k\alpha_k)\pi_k(t) - r_k\beta_k\pi_{k-1}(t)$$

$$(t-z)t\hat{\pi}_k(t) = (t-z)[\hat{\pi}_{k+1} + \hat{\alpha}_k\hat{\pi}_k(t) + \hat{\beta}_k\hat{\pi}_{k-1}(t)]$$

$$= \pi_{k+2}(t) + (\hat{\alpha}_k - r_{k+1})\pi_{k+1}(t) + (\hat{\beta}_k - r_k\hat{\alpha}_k)\pi_k(t) - r_{k-1}\hat{\beta}_k\pi_{k-1}(t)$$

comparison:

$$\hat{\alpha}_k - r_{k+1} = \alpha_{k+1} - r_k, \quad r_{k-1}\hat{\beta}_k = r_k\beta_k$$

Algorithm (Modification by a linear factor $t - z$)

initialization:

$$r_0 = z - \alpha_0, \quad r_1 = z - \alpha_1 - \beta_1 / r_0,$$
$$\hat{\alpha}_0 = \alpha_1 + r_1 - r_0, \quad \hat{\beta}_0 = -r_0 \beta_0$$

Algorithm **(Modification by a linear factor $t - z$)**

initialization:

$$r_0 = z - \alpha_0, \quad r_1 = z - \alpha_1 - \beta_1/r_0,$$
$$\hat{\alpha}_0 = \alpha_1 + r_1 - r_0, \quad \hat{\beta}_0 = -r_0\,\beta_0$$

continuation (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$r_{k+1} = z - \alpha_{k+1} - \beta_{k+1}/r_k,$$
$$\hat{\alpha}_k = \alpha_{k+1} + r_{k+1} - r_k,$$
$$\hat{\beta}_k = \beta_k r_k / r_{k-1}$$

Algorithm **(Modification by a linear factor $t - z$)**

initialization:

$$r_0 = z - \alpha_0, \quad r_1 = z - \alpha_1 - \beta_1/r_0,$$
$$\hat{\alpha}_0 = \alpha_1 + r_1 - r_0, \quad \hat{\beta}_0 = -r_0\,\beta_0$$

continuation (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$r_{k+1} = z - \alpha_{k+1} - \beta_{k+1}/r_k,$$
$$\hat{\alpha}_k = \alpha_{k+1} + r_{k+1} - r_k,$$
$$\hat{\beta}_k = \beta_k r_k/r_{k-1}$$

Matlab

```
ab=chri1(N,ab0,z)
```

Algorithm **(quadratic factor $(t-x)^2+y^2 = (t-z)(t-\overline{z})$)**

initialization:

$$r_0 = z - \alpha_0,\ r_1 = z - \alpha_1 - \beta_1/r_0,\ r_2 = z - \alpha_2 - \beta_2/r_1,$$

$$\hat{\alpha}_0 = \alpha_2 + r_2' + \frac{r_2''}{r_1''}\, r_1' - \left(r_1' + \frac{r_1''}{r_0''}\, r_0' \right),$$

$$\hat{\beta}_0 = \beta_0(\beta_1 + |r_0|^2)$$

Algorithm (quadratic factor $(t-x)^2+y^2=(t-z)(t-\overline{z})$)

initialization:

$$r_0 = z - \alpha_0,\ r_1 = z - \alpha_1 - \beta_1/r_0,\ r_2 = z - \alpha_2 - \beta_2/r_1,$$

$$\hat{\alpha}_0 = \alpha_2 + r_2' + \frac{r_2''}{r_1''}\, r_1' - \left(r_1' + \frac{r_1''}{r_0''}\, r_0' \right),$$

$$\hat{\beta}_0 = \beta_0(\beta_1 + |r_0|^2)$$

continuation (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$r_{k+2} = z - \alpha_{k+2} - \beta_{k+2}/r_{k+1},$$

$$\hat{\alpha}_k = \alpha_{k+2} + r_{k+2}' + \frac{r_{k+2}''}{r_{k+1}''}\, r_{k+1}' - \left(r_{k+1}' + \frac{r_{k+1}''}{r_k''}\, r_k' \right),$$

$$\hat{\beta}_k = \beta_k \frac{r_{k+1}'' r_{k-1}''}{[r_k'']^2} \left| \frac{r_k}{r_{k-1}} \right|^2$$

Algorithm **(quadratic factor $(t-x)^2+y^2=(t-z)(t-\overline{z})$)**

initialization:

$$r_0 = z-\alpha_0,\ r_1 = z-\alpha_1-\beta_1/r_0,\ r_2 = z-\alpha_2-\beta_2/r_1,$$

$$\hat{\alpha}_0 = \alpha_2 + r_2' + \frac{r_2''}{r_1''}\,r_1' - \left(r_1' + \frac{r_1''}{r_0''}\,r_0'\right),$$

$$\hat{\beta}_0 = \beta_0(\beta_1 + |r_0|^2)$$

continuation (if $n>1$): for $k=1,2,\dots,n-1$ do

$$r_{k+2} = z-\alpha_{k+2}-\beta_{k+2}/r_{k+1},$$

$$\hat{\alpha}_k = \alpha_{k+2} + r_{k+2}' + \frac{r_{k+2}''}{r_{k+1}''}\,r_{k+1}' - \left(r_{k+1}' + \frac{r_{k+1}''}{r_k''}\,r_k'\right),$$

$$\hat{\beta}_k = \beta_k \frac{r_{k+1}'' r_{k-1}''}{[r_k'']^2}\left|\frac{r_k}{r_{k-1}}\right|^2$$

Matlab `ab=chri2(N,ab0,x,y)`

linear divisor $\mathrm{d}\hat{\lambda}(t) = \frac{\mathrm{d}\lambda(t)}{t-z},\ z \in \mathbb{C}\backslash[a,b]$

generalized Christoffel:

$$\hat{\pi}_n(t) = \frac{\begin{vmatrix} \pi_{n-1}(t) & \pi_n(t) \\ \rho_{n-1}(z) & \rho_n(z) \end{vmatrix}}{-\rho_{n-1}(z)} = \pi_n(t) - r_{n-1}\pi_{n-1}(t), \quad r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)}$$

linear divisor $\mathrm{d}\hat{\lambda}(t) = \frac{\mathrm{d}\lambda(t)}{t-z},\ z \in \mathbb{C}\backslash[a,b]$

generalized Christoffel:

$$\hat{\pi}_n(t) = \frac{\begin{vmatrix} \pi_{n-1}(t) & \pi_n(t) \\ \rho_{n-1}(z) & \rho_n(z) \end{vmatrix}}{-\rho_{n-1}(z)} = \pi_n(t) - r_{n-1}\pi_{n-1}(t), \quad r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)}$$

Algorithm (Modifi cation by a linear divisor $t - z$)

initialization:

$$\hat{\alpha}_0 = \alpha_0 + r_0, \quad \hat{\beta}_0 = -\rho_0(z)$$

continuation (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$\hat{\alpha}_k = \alpha_k + r_k - r_{k-1},$$

$$\hat{\beta}_k = \beta_{k-1} r_{k-1} / r_{k-2}$$

computation of the r_k: continued fraction algorithm

linear divisor $d\hat{\lambda}(t) = \frac{d\lambda(t)}{t-z}, \; z \in \mathbb{C}\backslash[a,b]$

generalized Christoffel:

$$\hat{\pi}_n(t) = \frac{\begin{vmatrix} \pi_{n-1}(t) & \pi_n(t) \\ \rho_{n-1}(z) & \rho_n(z) \end{vmatrix}}{-\rho_{n-1}(z)} = \pi_n(t) - r_{n-1}\pi_{n-1}(t), \quad r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)}$$

Algorithm (Modifi cation by a linear divisor $t - z$)

initialization:

$$\hat{\alpha}_0 = \alpha_0 + r_0, \quad \hat{\beta}_0 = -\rho_0(z)$$

continuation (if $n > 1$): for $k = 1, 2, \ldots, n-1$ do

$$\hat{\alpha}_k = \alpha_k + r_k - r_{k-1},$$

$$\hat{\beta}_k = \beta_{k-1} r_{k-1} / r_{k-2}$$

computation of the r_k: continued fraction algorithm

Matlab

```
[ab,nu]=chri4(N,ab0,z,eps0,nu0,numax,rho0,iopt)
```

quadratic divisor $\mathrm{d}\hat{\lambda}(t) = \frac{\mathrm{d}\lambda(t)}{(t-x)^2+y^2}$

notations

$$r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)} = r'_n + \mathrm{i}r''_n$$

$$s_n = -\left(r'_{n-1} + \frac{r''_{n-1}}{r''_{n-2}}\, r'_{n-2}\right),\ n \geq 1; \quad t_n = \frac{r''_{n-1}}{r''_{n-2}}\,|r_{n-2}|^2,\ n \geq 2$$

quadratic divisor $\mathrm{d}\hat{\lambda}(t) = \frac{\mathrm{d}\lambda(t)}{(t-x)^2+y^2}$

notations

$$r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)} = r_n' + \mathrm{i} r_n''$$

$$s_n = -\left(r_{n-1}' + \frac{r_{n-1}''}{r_{n-2}''}\, r_{n-2}'\right),\; n \geq 1; \quad t_n = \frac{r_{n-1}''}{r_{n-2}''}\,|r_{n-2}|^2,\; n \geq 2$$

Algorithm

initialization:

$$\hat{\alpha}_0 = x + \rho_0' y/\rho_0'', \quad \hat{\beta}_0 = -\rho_0''/y,$$

$$\hat{\alpha}_1 = \alpha_1 - s_2 + s_1, \quad \hat{\beta}_1 = \beta_1 + s_1(\alpha_0 - \hat{\alpha}_1) - t_2,$$

$$\hat{\alpha}_2 = \alpha_2 - s_3 + s_2, \quad \hat{\beta}_2 = \beta_2 + s_2(\alpha_1 - \hat{\alpha}_2) - t_3 + t_2$$

quadratic divisor $\mathrm{d}\hat{\lambda}(t) = \frac{\mathrm{d}\lambda(t)}{(t-x)^2+y^2}$

notations

$$r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)} = r'_n + \mathrm{i}r''_n$$

$$s_n = -\left(r'_{n-1} + \frac{r''_{n-1}}{r''_{n-2}}\, r'_{n-2}\right),\ n \geq 1; \quad t_n = \frac{r''_{n-1}}{r''_{n-2}}\, |r_{n-2}|^2,\ n \geq 2$$

Algorithm

initialization:

$$\hat{\alpha}_0 = x + \rho'_0 y/\rho''_0, \quad \hat{\beta}_0 = -\rho''_0/y,$$

$$\hat{\alpha}_1 = \alpha_1 - s_2 + s_1, \quad \hat{\beta}_1 = \beta_1 + s_1(\alpha_0 - \hat{\alpha}_1) - t_2,$$

$$\hat{\alpha}_2 = \alpha_2 - s_3 + s_2, \quad \hat{\beta}_2 = \beta_2 + s_2(\alpha_1 - \hat{\alpha}_2) - t_3 + t_2$$

continuation (if $n > 3$): for $k = 3, 4, \ldots, n-1$ do

$$\hat{\alpha}_k = \alpha_k - s_{k+1} + s_k, \quad \hat{\beta}_k = \beta_{k-2}\, t_k/t_{k-1}$$

quadratic divisor $\mathrm{d}\hat{\lambda}(t) = \frac{\mathrm{d}\lambda(t)}{(t-x)^2+y^2}$

notations

$$r_n = \frac{\rho_{n+1}(z)}{\rho_n(z)} = r'_n + \mathrm{i}r''_n$$

$$s_n = -\left(r'_{n-1} + \frac{r''_{n-1}}{r''_{n-2}}\, r'_{n-2}\right),\ n \geq 1; \quad t_n = \frac{r''_{n-1}}{r''_{n-2}}\, |r_{n-2}|^2,\ n \geq 2$$

Algorithm

initialization:

$$\hat{\alpha}_0 = x + \rho'_0 y/\rho''_0, \quad \hat{\beta}_0 = -\rho''_0/y,$$

$$\hat{\alpha}_1 = \alpha_1 - s_2 + s_1, \quad \hat{\beta}_1 = \beta_1 + s_1(\alpha_0 - \hat{\alpha}_1) - t_2,$$

$$\hat{\alpha}_2 = \alpha_2 - s_3 + s_2, \quad \hat{\beta}_2 = \beta_2 + s_2(\alpha_1 - \hat{\alpha}_2) - t_3 + t_2$$

continuation (if $n > 3$): for $k = 3, 4, \ldots, n-1$ do

$$\hat{\alpha}_k = \alpha_k - s_{k+1} + s_k, \quad \hat{\beta}_k = \beta_{k-2}\, t_k/t_{k-1}$$

Matlab

```
[ab,nu]=chri5(N,ab0,z,eps0,nu0,numax,rho0,iopt)
```

Sobolev orthogonal polynomials

Sobolev orthogonal polynomials

Sobolev inner product

$$(p,q)_S = \int_{\mathbb{R}} p(t)q(t)\mathrm{d}\lambda_0(t) + \int_{\mathbb{R}} p'(t)q'(t)\mathrm{d}\lambda_1(t)$$

$$+ \cdots + \int_{\mathbb{R}} p^{(s)}(t)q^{(s)}(t)\mathrm{d}\lambda_s(t)$$

Sobolev orthogonal polynomials

Sobolev inner product

$$(p,q)_S = \int_{\mathbb{R}} p(t)q(t)\mathrm{d}\lambda_0(t) + \int_{\mathbb{R}} p'(t)q'(t)\mathrm{d}\lambda_1(t)$$

$$+ \cdots + \int_{\mathbb{R}} p^{(s)}(t)q^{(s)}(t)\mathrm{d}\lambda_s(t)$$

Sobolev orthogonal polynomials $\{\pi_k(\,\cdot\,;S)\}$

$$(\pi_k, \pi_\ell)_S \begin{cases} = 0, & k \neq \ell \\ > 0, & k = \ell \end{cases}$$

Sobolev orthogonal polynomials

Sobolev inner product

$$(p,q)_S = \int_{\mathbb{R}} p(t)q(t)\mathrm{d}\lambda_0(t) + \int_{\mathbb{R}} p'(t)q'(t)\mathrm{d}\lambda_1(t)$$

$$+ \cdots + \int_{\mathbb{R}} p^{(s)}(t)q^{(s)}(t)\mathrm{d}\lambda_s(t)$$

Sobolev orthogonal polynomials $\{\pi_k(\,\cdot\,;S)\}$

$$(\pi_k,\pi_\ell)_S \begin{cases} =0, & k\neq\ell \\ >0, & k=\ell \end{cases}$$

recurrence relation

$$\pi_{k+1}(t) = t\pi_k(t) - \sum_{j=0}^{k} \beta_j^k \pi_{k-j}(t), \quad k=0,1,2,\ldots$$

Recurrence matrix

recurrence relation

$$\pi_{k+1}(t) = t\pi_k(t) - \sum_{j=0}^{k} \beta_j^k \pi_{k-j}(t),\ k = 0, 1, 2, \ldots$$

matrix of recurrence coefficients

$$\boldsymbol{H}_n = \begin{bmatrix} \beta_0^0 & \beta_1^1 & \beta_2^2 & \cdots & \beta_{n-2}^{n-2} & \beta_{n-1}^{n-1} \\ 1 & \beta_0^1 & \beta_1^2 & \cdots & \beta_{n-3}^{n-2} & \beta_{n-2}^{n-1} \\ 0 & 1 & \beta_0^2 & \cdots & \beta_{n-4}^{n-2} & \beta_{n-3}^{n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \beta_0^{n-2} & \beta_1^{n-1} \\ 0 & 0 & 0 & \cdots & 1 & \beta_0^{n-1} \end{bmatrix}$$

Modified Chebyshev algorithm

Modified Chebyshev algorithm

modified moments

$$m_k^{(\sigma)} = \int_{\mathbb{R}} p_k(t) \mathrm{d}\lambda_\sigma,$$
$$k = 0, 1, 2, \ldots, ; \;\; \sigma = 0, 1, \ldots, s$$

Modified Chebyshev algorithm

modified moments

$$m_k^{(\sigma)} = \int_{\mathbb{R}} p_k(t) \mathrm{d}\lambda_\sigma,$$
$$k = 0, 1, 2, \ldots, ; \;\; \sigma = 0, 1, \ldots, s$$

modified moment map

$$[m_k^{(\sigma)}]_{k=0}^{2n-1}, \; \sigma = 0, 1, \ldots, s \; \mapsto \boldsymbol{H}_n$$

Modified Chebyshev algorithm

modified moments

$$m_k^{(\sigma)} = \int_{\mathbb{R}} p_k(t)\mathrm{d}\lambda_\sigma,$$
$$k = 0, 1, 2, \ldots, ; \;\; \sigma = 0, 1, \ldots, s$$

modified moment map

$$[m_k^{(\sigma)}]_{k=0}^{2n-1},\ \sigma = 0, 1, \ldots, s \ \mapsto \boldsymbol{H}_n$$

algorithm G. and Zhang, 1995

conditioning Zhang, 1994

Matlab (for $s = 1$)

```
B=chebyshev_sob(N,mom,abm)
```

Example $\mathrm{d}\lambda_0(t) = \mathrm{d}t, \quad \mathrm{d}\lambda_1(t) = \gamma \mathrm{d}t$ on $[-1, 1]$
(Althammer, 1962)

modified moments

$$p_k(t) = \text{monic Legendre}$$
$$m_0^{(0)} = 2,\ m_0^{(1)} = 2\gamma; \quad m_k^{(0)} = m_k^{(1)} = 0,\ k > 0$$

Matlab

```
mom=zeros(2,2*N);
mom(1,1)=2;   mom(2,1)=2*g;
abm=r_jacobi(2*N-1);
B=chebyshev_sob(N,mom,abm);
```

Discretized Stieltjes algorithm

Discretized Stieltjes algorithm

(G. and Zhang, 1995)

$$\beta_j^k = \frac{(t\pi_k, \pi_{k-j})_S}{(\pi_{k-j}, \pi_{k-j})_S}, \quad j = 0, 1, \ldots, k$$

Discretized Stieltjes algorithm

(G. and Zhang, 1995)

$$\beta_j^k = \frac{(t\pi_k, \pi_{k-j})_S}{(\pi_{k-j}, \pi_{k-j})_S}, \quad j = 0, 1, \ldots, k$$

Gauss quadrature discretization

$$(p, q)_{d\lambda_\sigma} \approx \sum_{k=1}^{n_\sigma} w_k^{(\sigma)} p(x_k^{(\sigma)}) q(x_k^{(\sigma)}), \;\; \sigma = 0, 1, \ldots, s$$

Matlab

`B=stieltjes_sob(N,s,nd,xw,a0,same)`

`a0=`$\alpha_0(d\lambda_0)$

Matlab (cont')

$$
\texttt{xw=}\left\|\begin{array}{ccc|ccc}
x_1^{(0)} & \cdots & x_1^{(s)} & w_1^{(0)} & \cdots & w_1^{(s)} \\
x_2^{(0)} & \cdots & x_2^{(s)} & w_2^{(0)} & \cdots & w_2^{(s)} \\
\vdots & & \vdots & \vdots & & \vdots \\
x_{\mathrm{md}}^{(0)} & \cdots & x_{\mathrm{md}}^{(s)} & w_{\mathrm{md}}^{(0)} & \cdots & w_{\mathrm{md}}^{(s)}
\end{array}\right\|
$$

`md=max(nd)`

Example Althammer’s polynomials

Matlab

```
s=1; nd=[N N];
a0=0; same=1;
ab=r_jacobi(N);
zw=gauss(N,ab);
xw=[zw(:,1) zw(:,1) ...
     zw(:,2) g*zw(:,2)];
B=stieltjes_sob(N,s,nd,xw,a0,same);
```

Zeros

if

$$\boldsymbol{\pi}^T(t) = [\pi_0(t), \pi_1(t), \ldots, \pi_{n-1}(t)]$$

then

$$t\boldsymbol{\pi}^T(t) = \boldsymbol{\pi}^T(t)\boldsymbol{H}_n + \pi_n(t)\boldsymbol{e}_n^T$$

Theorem   The zeros τ_ν of π_n are the eigenvalues of $\boldsymbol{H}_n$ and $\boldsymbol{\pi}^T(\tau_\nu)$ corresponding (left) eigenvectors. $\square$

Matlab

```
z=sobzeros(n,N,B)
```

Part II

Quadrature

Gauss quadrature formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G) + R_n^G(f)$$

Gauss quadrature formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G) + R_n^G(f)$$

Theorem (Golub and Welsch, 1969)
The nodes τ_ν^G are the eigenvalues of $\boldsymbol{J}_n(\mathrm{d}\lambda)$ and the weights λ_ν^G are

$$\lambda_\nu^G = \beta_0 \boldsymbol{v}_{\nu,1}^2$$

where $\beta_0 = \int_{\mathbb{R}} \mathrm{d}\lambda(t)$ and $\boldsymbol{v}_{\nu,1}$ is the first component of the normalized eigenvector $\boldsymbol{v}_\nu$,

$$\boldsymbol{J}_n(\mathrm{d}\lambda)\boldsymbol{v}_\nu = \lambda_\nu^G \boldsymbol{v}_\nu, \quad \boldsymbol{v}_\nu^T \boldsymbol{v}_\nu = 1. \ \square$$

Gauss quadrature formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G) + R_n^G(f)$$

Theorem (Golub and Welsch, 1969)
The nodes τ_ν^G are the eigenvalues of $\boldsymbol{J}_n(\mathrm{d}\lambda)$ and the weights λ_ν^G are

$$\lambda_\nu^G = \beta_0 \boldsymbol{v}_{\nu,1}^2$$

where $\beta_0 = \int_{\mathbb{R}} \mathrm{d}\lambda(t)$ and $\boldsymbol{v}_{\nu,1}$ is the first component of the normalized eigenvector $\boldsymbol{v}_\nu$,

$$\boldsymbol{J}_n(\mathrm{d}\lambda)\boldsymbol{v}_\nu = \lambda_\nu^G \boldsymbol{v}_\nu, \quad \boldsymbol{v}_\nu^T \boldsymbol{v}_\nu = 1. \ \square$$

Matlab

```
xw=gauss(N,ab)
```

Corollary If f is sufficiently smooth, then

$$\sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G) = \beta_0 \boldsymbol{e}_1^T f(\boldsymbol{J}_n(\mathrm{d}\lambda))\boldsymbol{e}_1, \; \boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$$

Corollary If f is sufficiently smooth, then

$$\sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G) = \beta_0 \boldsymbol{e}_1^T f(\boldsymbol{J}_n(\mathrm{d}\lambda))\boldsymbol{e}_1, \ \boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$$

Proof Let $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$ and $\boldsymbol{J} = \boldsymbol{J}_n(\mathrm{d}\lambda)$. Then (spectral decomposition of $\boldsymbol{J}$)

$$\boldsymbol{J}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}_\tau, \quad \boldsymbol{D}_\tau = \mathrm{diag}(\tau_1^G, \tau_2^G, \ldots, \tau_n^G)$$

and

$$\begin{aligned}\beta_0 \boldsymbol{e}_1^T f(\boldsymbol{J})\boldsymbol{e}_1 &= \beta_0 \boldsymbol{e}_1^T \boldsymbol{V} f(\boldsymbol{D}_\tau)\boldsymbol{V}^T \boldsymbol{e}_1 \\ &= \beta_0 \sum_{\nu=1}^{n} \boldsymbol{v}_{\nu,1}^2 f(\tau_\nu^G) = \sum_{\nu=1}^{n} \lambda_\nu^G f(\tau_\nu^G)\end{aligned}$$

since $\beta_0 \boldsymbol{v}_{\nu,1}^2 = \lambda_\nu^G$. $\square$

Example Zeros of Sobolev orthogonal polynomials of Gegenbauer type (Groenevelt, 2002)

Sobolev inner product

$$(u,v)_S = \int_{-1}^{1} u(t)v(t)(1-t^2)^{\alpha-1}\mathrm{d}t + \gamma \int_{-1}^{1} u'(t)v'(t)\frac{(1-t^2)^{\alpha}}{t^2+y^2}\,\mathrm{d}t$$

Example Zeros of Sobolev orthogonal polynomials of Gegenbauer type (Groenevelt, 2002)

Sobolev inner product

$$(u,v)_S = \int_{-1}^{1} u(t)v(t)(1-t^2)^{\alpha-1}\mathrm{d}t + \gamma \int_{-1}^{1} u'(t)v'(t)\frac{(1-t^2)^{\alpha}}{t^2+y^2}\,\mathrm{d}t$$

Matlab program

```
s=1; same=0; eps0=1e-14; numax=250; nd=[N N];
ab0=r_jacobi(numax,alpha);
z=complex(0,y);
nu0=nu0jac(N,z,eps0); rho0=0; iopt=1;
ab1=chri6(N,ab0,y,eps0,nu0,numax,rho0,iopt);
zw1=gauss(N,ab1);
ab=r_jacobi(N,alpha-1); zw=gauss(N,ab);
xw=[zw(:,1) zw1(:,1) zw(:,2) gamma*zw1(:,2)];
a0=ab(1,1); B=stieltjes_sob(N,s,nd,xw,a0,same)
z=sobzeros(N,N,B)
```

Demo #4 N=12, $\alpha = \frac{1}{2}$, $\gamma = 1$

Demo #4 N=12, $\alpha = \frac{1}{2}$, $\gamma = 1$

y	zeros	y	zeros
.1	.027543282225	.09	.011086169153 i
	.284410786673		.281480077515
	.541878443180		.540697645595
	.756375307278		.755863108617
	.909868274113		.909697039063
	.989848649239		.989830182743

All other zeros are the same with opposite signs

Gauss-Radau formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \lambda_0^a f(a) + \sum_{\nu=1}^{n} \lambda_\nu^a f(\tau_\nu^a) + R_n^a(f)$$

Gauss-Radau formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \lambda_0^a f(a) + \sum_{\nu=1}^{n} \lambda_\nu^a f(\tau_\nu^a) + R_n^a(f)$$

Jacobi-Radau matrix

$$\boldsymbol{J}_{n+1}^{R,a}(\mathrm{d}\lambda) = \begin{bmatrix} \boldsymbol{J}_n(\mathrm{d}\lambda) & \sqrt{\beta_n}\boldsymbol{e}_n \\ \sqrt{\beta_n}\boldsymbol{e}_n^T & \alpha_n^R \end{bmatrix}, \ \alpha_n^R = a - \beta_n \frac{\pi_{n-1}(a)}{\pi_n(a)}$$

Jacobi resp. Laguerre measure

$$\alpha_n^R = -1 + \frac{2n(n+\alpha)}{(2n+\alpha+\beta)(2n+\alpha+\beta+1)}, \quad \alpha_n^R = n$$

Gauss-Radau formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \lambda_0^a f(a) + \sum_{\nu=1}^n \lambda_\nu^a f(\tau_\nu^a) + R_n^a(f)$$

Jacobi-Radau matrix

$$\boldsymbol{J}_{n+1}^{R,a}(\mathrm{d}\lambda) = \begin{bmatrix} \boldsymbol{J}_n(\mathrm{d}\lambda) & \sqrt{\beta_n}\boldsymbol{e}_n \\ \sqrt{\beta_n}\boldsymbol{e}_n^T & \alpha_n^R \end{bmatrix}, \; \alpha_n^R = a - \beta_n \frac{\pi_{n-1}(a)}{\pi_n(a)}$$

Jacobi resp. Laguerre measure

$$\alpha_n^R = -1 + \frac{2n(n+\alpha)}{(2n+\alpha+\beta)(2n+\alpha+\beta+1)}, \quad \alpha_n^R = n$$

Matlab

```
xw=radau(N,ab,end0)
xw=radau_jacobi(N,iopt,a,b)
xw=radau_laguerre(N,a)
```

Gauss-Lobatto formula

$$\int_a^b f(t)\mathrm{d}\lambda(t) \approx \lambda_0^L f(a) + \sum_{\nu=1}^n \lambda_\nu^L f(\tau_\nu^L) + \lambda_{n+1}^L f(b)$$

Gauss-Lobatto formula

$$\int_a^b f(t)\mathrm{d}\lambda(t) \approx \lambda_0^L f(a) + \sum_{\nu=1}^n \lambda_\nu^L f(\tau_\nu^L) + \lambda_{n+1}^L f(b)$$

Jacobi-Lobatto matrix

$$\boldsymbol{J}_{n+2}^L(\mathrm{d}\lambda) = \begin{bmatrix} \boldsymbol{J}_{n+1}(\mathrm{d}\lambda) & \sqrt{\beta_{n+1}^L}\boldsymbol{e}_{n+1} \\ \sqrt{\beta_{n+1}^L}\boldsymbol{e}_{n+1}^T & \alpha_{n+1}^L \end{bmatrix}$$

$$\begin{bmatrix} \pi_{n+1}(a) & \pi_n(a) \\ \pi_{n+1}(b) & \pi_n(b) \end{bmatrix} \begin{bmatrix} \alpha_{n+1}^L \\ \beta_{n+1}^L \end{bmatrix} = \begin{bmatrix} a\pi_{n+1}(a) \\ b\pi_{n+1}(b) \end{bmatrix}$$

Jacobi measure

$$\alpha_{n+1}^L = \frac{\alpha-\beta}{2n+\alpha+\beta+2}, \; \beta_{n+1}^L = 4\frac{(n+\alpha+1)(n+\beta+1)(n+\alpha+\beta+1)}{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}$$

Gauss-Lobatto formula

$$\int_a^b f(t)\mathrm{d}\lambda(t) \approx \lambda_0^L f(a) + \sum_{\nu=1}^n \lambda_\nu^L f(\tau_\nu^L) + \lambda_{n+1}^L f(b)$$

Jacobi-Lobatto matrix

$$\boldsymbol{J}_{n+2}^L(\mathrm{d}\lambda) = \begin{bmatrix} \boldsymbol{J}_{n+1}(\mathrm{d}\lambda) & \sqrt{\beta_{n+1}^L}\boldsymbol{e}_{n+1} \\ \sqrt{\beta_{n+1}^L}\boldsymbol{e}_{n+1}^T & \alpha_{n+1}^L \end{bmatrix}$$

$$\begin{bmatrix} \pi_{n+1}(a) & \pi_n(a) \\ \pi_{n+1}(b) & \pi_n(b) \end{bmatrix} \begin{bmatrix} \alpha_{n+1}^L \\ \beta_{n+1}^L \end{bmatrix} = \begin{bmatrix} a\pi_{n+1}(a) \\ b\pi_{n+1}(b) \end{bmatrix}$$

Jacobi measure

$$\alpha_{n+1}^L = \frac{\alpha-\beta}{2n+\alpha+\beta+2}, \quad \beta_{n+1}^L = 4\frac{(n+\alpha+1)(n+\beta+1)(n+\alpha+\beta+1)}{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}$$

Matlab

```
xw=lobatto(N,ab,endl,endr)
xw=lobatto_jacobi(N,a,b)
```

Error bounds

Gauss-Radau formulae

$$\int_a^b f(t)\mathrm{d}\lambda(t) = \lambda_0^a f(a) + \sum_{\nu=1}^n \lambda_\nu^a f(\tau_\nu^a) + R_n^a(f)$$
$$= \sum_{\nu=1}^n \lambda_\nu^b f(\tau_\nu^b) + \lambda_{n+1}^b f(b) + R_n^b(f)$$

Theorem If $f \in C^{2n+1}[a,b]$, then

$$R_n^a(f) > 0,\ R_n^b(f) < 0 \quad \text{if sgn } f^{(2n+1)} = 1 \text{ on } [a,b],$$

with the inequalities reversed if sgn $f^{(2n+1)} = -1$.

Error bounds

Gauss-Radau formulae

$$\int_a^b f(t)\mathrm{d}\lambda(t) = \lambda_0^a f(a) + \sum_{\nu=1}^n \lambda_\nu^a f(\tau_\nu^a) + R_n^a(f)$$
$$= \sum_{\nu=1}^n \lambda_\nu^b f(\tau_\nu^b) + \lambda_{n+1}^b f(b) + R_n^b(f)$$

Theorem If $f \in C^{2n+1}[a,b]$, then

$$R_n^a(f) > 0,\ R_n^b(f) < 0 \quad \text{if sgn } f^{(2n+1)} = 1 \text{ on } [a,b],$$

with the inequalities reversed if sgn $f^{(2n+1)} = -1$.

Proof (for R_n^a; for R_n^b the proof is similar)

$$R_n^a(f) = \gamma_n^a \frac{f^{(2n+1)}(\tau^a)}{(2n+1)!},\ \gamma_n^a = \int_a^b [\pi_n(t;\mathrm{d}\lambda_a]^2 \mathrm{d}\lambda_a(t),$$

where $\mathrm{d}\lambda_a(t) = (t-a)\mathrm{d}\lambda(t)$. $\square$

Gauss-Lobatto formula

$$\int_a^b f(t)\mathrm{d}\lambda(t) = \lambda_0^L f(a) + \sum_{\nu=1}^n \lambda_\nu^L f(\tau_\nu^L) + \lambda_{n+1} f(b) + R_n^{a,b}(f)$$

Theorem If $f \in C^{2n+2}[a,b]$, then

$$R_n^{a,b}(f) < 0 \quad \text{if } \operatorname{sgn} f^{(2n+2)} = 1 \text{ on } [a,b],$$

with the inequality reversed if $\operatorname{sgn} f^{(2n+2)} = -1$.

Gauss-Lobatto formula

$$\int_a^b f(t)\mathrm{d}\lambda(t) = \lambda_0^L f(a) + \sum_{\nu=1}^n \lambda_\nu^L f(\tau_\nu^L) + \lambda_{n+1} f(b) + R_n^{a,b}(f)$$

Theorem   If $f \in C^{2n+2}[a,b]$, then

$$R_n^{a,b}(f) < 0 \quad \text{if sgn } f^{(2n+2)} = 1 \text{ on } [a,b],$$

with the inequality reversed if sgn $f^{(2n+2)} = -1$.

Proof

$$R_n^{a,b}(f) = -\gamma_n \frac{f^{(2n+2)}(\tau)}{(2n+2)!},\ \gamma_n = \int_a^b [\pi_n(t;\mathrm{d}\lambda_{a,b})]^2 \mathrm{d}\lambda_{a,b}(t),$$

where $\mathrm{d}\lambda_{a,b}(t) = (t-a)(b-t)\mathrm{d}\lambda(t)$. $\square$

Gauss-Kronrod formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n} \lambda_\nu^K f(\tau_\nu^G) + \sum_{\mu=1}^{n+1} \lambda_\mu^{*K} f(\tau_\mu^K)$$

Gauss-Kronrod formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n} \lambda_\nu^K f(\tau_\nu^G) + \sum_{\mu=1}^{n+1} \lambda_\mu^{*K} f(\tau_\mu^K)$$

Jacobi-Kronrod matrix (Laurie, 1997)

$$\boldsymbol{J}_{2n+1}^K(\mathrm{d}\lambda) = \begin{bmatrix} \boldsymbol{J}_n(\mathrm{d}\lambda) & \sqrt{\beta_n}\boldsymbol{e}_n & \boldsymbol{0} \\ \sqrt{\beta_n}\boldsymbol{e}_n^T & \alpha_n & \sqrt{\beta_{n+1}}\boldsymbol{e}_1^T \\ \boldsymbol{0} & \sqrt{\beta_{n+1}}\boldsymbol{e}_1 & \boldsymbol{J}_n^* \end{bmatrix}$$

The trailing (symmetric, tridiagonal, and partially known) matrix $\boldsymbol{J}_n^*$ can be computed by Laurie's Algorithm

Gauss-Kronrod formula

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n} \lambda_\nu^K f(\tau_\nu^G) + \sum_{\mu=1}^{n+1} \lambda_\mu^{*K} f(\tau_\mu^K)$$

Jacobi-Kronrod matrix (Laurie, 1997)

$$\boldsymbol{J}_{2n+1}^K(\mathrm{d}\lambda) = \begin{bmatrix} \boldsymbol{J}_n(\mathrm{d}\lambda) & \sqrt{\beta_n}\boldsymbol{e}_n & \boldsymbol{0} \\ \sqrt{\beta_n}\boldsymbol{e}_n^T & \alpha_n & \sqrt{\beta_{n+1}}\boldsymbol{e}_1^T \\ \boldsymbol{0} & \sqrt{\beta_{n+1}}\boldsymbol{e}_1 & \boldsymbol{J}_n^* \end{bmatrix}$$

The trailing (symmetric, tridiagonal, and partially known) matrix $\boldsymbol{J}_n^*$ can be computed by Laurie's Algorithm

Matlab

```
ab=r_kronrod(N,ab0)
xw=kronrod(n,ab)
```

Gauss-Turán formula (Turán, 1950)

$$\int_{\mathbb{R}} f(t)\,\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n} [\lambda_\nu f(\tau_\nu) + \cdots + \lambda_\nu^{(r-1)} f^{(r-1)}(\tau_\nu)]$$

Gauss-Turán formula (Turán, 1950)

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n}[\lambda_\nu f(\tau_\nu) + \cdots + \lambda_\nu^{(r-1)} f^{(r-1)}(\tau_\nu)]$$

power orthogonality ($r = 2s+1,\, s \geq 0$)

$$\int_{\mathbb{R}} [\omega_n(t)]^{2s+1} p(t)\mathrm{d}\lambda(t) = 0, \quad \text{all } p \in \mathbb{P}_{n-1}$$

where $\omega_n(t) = \prod_{\nu=1}^{n}(t-\tau_\nu)$

s-orthogonal polynomial: $\omega_n(t) = \pi_{n,s}(t)$

computation (basic idea): $\mathrm{d}\lambda_{n,s}(t) := [\pi_{n,s}(t)]^{2s}\mathrm{d}\lambda(t)$

$$\int_R \pi_{n,s}(t)p(t)\mathrm{d}\lambda_{n,s}(t) = 0, \quad \text{all } p \in \mathbb{P}_{n-1}$$

Gauss-Turán formula (Turán, 1950)

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) \approx \sum_{\nu=1}^{n}[\lambda_\nu f(\tau_\nu) + \cdots + \lambda_\nu^{(r-1)} f^{(r-1)}(\tau_\nu)]$$

power orthogonality ($r = 2s+1,\ s \geq 0$)

$$\int_{\mathbb{R}}[\omega_n(t)]^{2s+1} p(t)\mathrm{d}\lambda(t) = 0, \quad \text{all } p \in \mathbb{P}_{n-1}$$

where $\omega_n(t) = \prod_{\nu=1}^{n}(t-\tau_\nu)$

s-orthogonal polynomial: $\omega_n(t) = \pi_{n,s}(t)$

computation (basic idea): $\mathrm{d}\lambda_{n,s}(t) := [\pi_{n,s}(t)]^{2s}\mathrm{d}\lambda(t)$

$$\int_R \pi_{n,s}(t) p(t)\mathrm{d}\lambda_{n,s}(t) = 0, \quad \text{all } p \in \mathbb{P}_{n-1}$$

Matlab

```
xw=turan(n,s,eps0,ab0,hom)
```

Polynomial/rational quadrature

$$\int_{\mathbb{R}} g(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n} \lambda_\nu g(\tau_\nu) + R_n(g)$$

Problem: determine λ_ν, τ_ν such that $R_n(g) = 0$ if $g \in \mathbb{S}_{2n}$, where

$$\mathbb{S}_{2n} = \mathbb{Q}_m \oplus \mathbb{P}_{2n-m-1}, \quad 0 \le m \le 2n$$

$$\mathbb{P}_{2n-m-1} = \text{polynomials of degree } \le 2n-m-1$$

$$\mathbb{Q}_m = \text{rational functions with prescribed poles}$$

Polynomial/rational quadrature

$$\int_{\mathbb{R}} g(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n} \lambda_\nu g(\tau_\nu) + R_n(g)$$

Problem: determine λ_ν, τ_ν such that $R_n(g) = 0$ if $g \in \mathbb{S}_{2n}$, where

$$\mathbb{S}_{2n} = \mathbb{Q}_m \oplus \mathbb{P}_{2n-m-1}, \quad 0 \le m \le 2n$$

$$\mathbb{P}_{2n-m-1} = \text{polynomials of degree } \le 2n-m-1$$

$$\mathbb{Q}_m = \text{rational functions with prescribed poles}$$

specifically:

$$\mathbb{Q}_m = \text{span}\left\{ r(t) = \tfrac{1}{1+\zeta_\mu t}, \;\; \mu = 1, 2, \ldots, m \right\}$$

$$\zeta_\mu \in \mathbb{C}, \quad \zeta_\mu \ne 0, \quad 1 + \zeta_\mu t \ne 0 \text{ on } \mathrm{supp}(\mathrm{d}\lambda)$$

Theorem (G., 2000; Van Assche et al., 2000)

Let $\omega_m(t)=\prod_{\mu=1}^m(1+\zeta_\mu t)$. Assume the existence of a (polynomial) Gauss formula

$$\int_{\mathbb{R}} g(t)\frac{\mathrm{d}\lambda(t)}{\omega_m(t)}=\sum_{\nu=1}^n \lambda_\nu^G g(\tau_\nu^G), \quad g\in\mathbb{P}_{2n-1}.$$

Then

$$\tau_\nu=\tau_\nu^G, \quad \lambda_\nu=\omega_m(\tau_\nu^G)\lambda_\nu^G, \quad \nu=1,2,\ldots,n,$$

yields the desired formula. $\square$

Theorem (G., 2000; Van Assche et al., 2000)

Let $\omega_m(t) = \prod_{\mu=1}^{m}(1+\zeta_\mu t)$. Assume the existence of a (polynomial) Gauss formula

$$\int_{\mathbb{R}} g(t)\frac{\mathrm{d}\lambda(t)}{\omega_m(t)} = \sum_{\nu=1}^{n} \lambda_\nu^G g(\tau_\nu^G), \quad g \in \mathbb{P}_{2n-1}.$$

Then

$$\tau_\nu = \tau_\nu^G, \quad \lambda_\nu = \omega_m(\tau_\nu^G)\lambda_\nu^G, \quad \nu = 1, 2, \ldots, n,$$

yields the desired formula. □

construction (assuming $\omega_m \geq 0$)

- discretization method using Gauss quadrature relative to $\mathrm{d}\lambda$
- special techniques for "difficult" poles

Example Fermi-Dirac integral $F_k(\eta, \theta) = \int_0^\infty \frac{t^k \sqrt{1+\theta t/2}}{e^{-\eta+t}+1}\, \mathrm{d}t$

$\eta \in \mathbb{R},\ \theta \geq 0,\ \ k \ldots$ Boltzmann constant ($= \frac{1}{2}, \frac{3}{2}$, or $\frac{5}{2}$)

equivalently

$$F_k(\eta, \theta) = \int_0^\infty \frac{\sqrt{1+\theta t/2}}{e^{-\eta}+e^{-t}}\, \mathrm{d}\lambda^{[k]}(t), \quad \mathrm{d}\lambda^{[k]}(t) = t^k e^{-t} \mathrm{d}t$$

Example Fermi-Dirac integral $F_k(\eta,\theta)=\int_0^\infty \frac{t^k\sqrt{1+\theta t/2}}{e^{-\eta+t}+1}\,\mathrm{d}t$

$\eta\in\mathbb{R},\ \theta\geq 0,\ \ k\ldots$ Boltzmann constant $(=\frac{1}{2},\frac{3}{2}$, or $\frac{5}{2})$

equivalently

$$F_k(\eta,\theta)=\int_0^\infty \frac{\sqrt{1+\theta t/2}}{e^{-\eta}+e^{-t}}\,\mathrm{d}\lambda^{[k]}(t),\qquad \mathrm{d}\lambda^{[k]}(t)=t^k e^{-t}\mathrm{d}t$$

poles at $\eta+\mu\mathrm{i}\pi,\mu=\pm1,\pm3,\pm5,\ldots$ (all "easy")

$$\omega_m(t)=\prod_{\nu=1}^{m/2}[(1+\xi_\nu t)^2+\eta_\nu t^2],\qquad 2\leq m(\text{even})\leq 2n$$

where

$$\xi_\nu=\frac{-\eta}{\eta^2+(2\nu-1)^2\pi^2}\,,\qquad \eta_\nu=\frac{(2\nu-1)\pi}{\eta^2+(2\nu-1)^2\pi^2}$$

Example Fermi-Dirac integral $F_k(\eta,\theta)=\int_0^\infty \frac{t^k\sqrt{1+\theta t/2}}{e^{-\eta+t}+1}\,\mathrm{d}t$

$\eta\in\mathbb{R},\ \theta\geq 0,\ \ k\ldots$ Boltzmann constant $(=\frac{1}{2},\frac{3}{2},$ or $\frac{5}{2})$

equivalently

$$F_k(\eta,\theta)=\int_0^\infty \frac{\sqrt{1+\theta t/2}}{e^{-\eta}+e^{-t}}\,\mathrm{d}\lambda^{[k]}(t),\qquad \mathrm{d}\lambda^{[k]}(t)=t^k e^{-t}\mathrm{d}t$$

poles at $\eta+\mu\mathrm{i}\pi,\mu=\pm1,\pm3,\pm5,\ldots$ (all "easy")

$$\omega_m(t)=\prod_{\nu=1}^{m/2}[(1+\xi_\nu t)^2+\eta_\nu t^2],\qquad 2\leq m(\text{even})\leq 2n$$

where

$$\xi_\nu=\frac{-\eta}{\eta^2+(2\nu-1)^2\pi^2}\,,\qquad \eta_\nu=\frac{(2\nu-1)\pi}{\eta^2+(2\nu-1)^2\pi^2}$$

rational Gauss approximation $F_k(\eta,\theta)\approx\sum_{n=1}^N w_n\frac{\sqrt{1+\theta x_n/2}}{e^{-\eta}+e^{-x_n}}$

Matlab

```
xw=fermi_dirac(N,m,eta,theta,k,eps0,Nmax)
```

Cauchy principal value integral

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = ⨍_a^b \frac{f(t)}{x-t}\,\mathrm{d}\lambda(t), \quad x \in (a,b)$$

Cauchy principal value integral

$$(\mathcal{C}f)(x; \mathrm{d}\lambda) = ⨍_a^b \frac{f(t)}{x-t}\, \mathrm{d}\lambda(t), \quad x \in (a,b)$$

modified quadrature rule

$$(\mathcal{C}f)(x; \mathrm{d}\lambda) = c_0(x) f(x) + \sum_{\nu=1}^{n} c_\nu(x) f(\tau_\nu) + R_n(f; x)$$

can be made "Gaussian": $R_n(\mathbb{P}_{2n}) = 0$

Cauchy principal value integral

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = \fint_a^b \frac{f(t)}{x-t}\,\mathrm{d}\lambda(t), \quad x \in (a,b)$$

modified quadrature rule

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = c_0(x)f(x) + \sum_{\nu=1}^{n} c_\nu(x)f(\tau_\nu) + R_n(f;x)$$

can be made "Gaussian": $R_n(\mathbb{P}_{2n}) = 0$
quadrature rule in the strict sense

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = \sum_{\nu=1}^{n} c_\nu^*(x)f(\tau_\nu^*) + R_n^*(f;x)$$

interpolatory at best: $R_n^*(\mathbb{P}_{n-1}) = 0$

Cauchy principal value integral

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = ⨍_a^b \frac{f(t)}{x-t}\,\mathrm{d}\lambda(t), \quad x \in (a,b)$$

modified quadrature rule

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = c_0(x)f(x) + \sum_{\nu=1}^{n} c_\nu(x)f(\tau_\nu) + R_n(f;x)$$

can be made "Gaussian": $R_n(\mathbb{P}_{2n}) = 0$
quadrature rule in the strict sense

$$(\mathcal{C}f)(x;\mathrm{d}\lambda) = \sum_{\nu=1}^{n} c_\nu^*(x)f(\tau_\nu^*) + R_n^*(f;x)$$

interpolatory at best: $R_n^*(\mathbb{P}_{n-1}) = 0$

Matlab

```
cpvi=cauchyPVI(N,x,f,ddf,iopt,ab,rho0)
```

`ddf(u,v)` $= \frac{f(u)-f(v)}{u-v}$, `rho0` $= ⨍_a^b \frac{\mathrm{d}\lambda(t)}{x-t}$

Polynomials orthogonal on several intervals

$$\mathrm{d}\lambda(t) = \sum_j \chi_{[c_j,d_j]}(t)\mathrm{d}\lambda_j(t)$$

Problem Given $\boldsymbol{J}^{(j)} = \boldsymbol{J}_n(\mathrm{d}\lambda_j)$, find $\boldsymbol{J} = \boldsymbol{J}_n(\mathrm{d}\lambda)$

Polynomials orthogonal on several intervals

$$d\lambda(t) = \sum_j \chi_{[c_j,d_j]}(t) d\lambda_j(t)$$

Problem Given $\boldsymbol{J}^{(j)} = \boldsymbol{J}_n(d\lambda_j)$, find $\boldsymbol{J} = \boldsymbol{J}_n(d\lambda)$

Solution by Stieltjes procedure

n-point Gauss quadrature discretization

$$\int_{c_j}^{d_j} p(t) d\lambda_j(t) = \sum_{\nu=1}^{n} \lambda_\nu^{(j)} p(\tau_\nu^{(j)}), \quad p \in \mathbb{P}_{2n-1}$$

by the corollary to the Golub-Welsch theorem

$$\int_{c_j}^{d_j} p(t) d\lambda_j(t) = \beta_0^{(j)} \boldsymbol{e}_1^T p(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1, \; \beta_0^{(j)} = \int_{c_j}^{d_j} d\lambda_j(t)$$

Solution (cont')

computation of the inner products ($k \le n-1$)

$$\begin{aligned}(t\pi_k, \pi_k)_{\mathrm{d}\lambda} &= \int_{\mathbb{R}} t\pi_k^2(t)\mathrm{d}\lambda(t) = \sum_j \int_{c_j}^{d_j} t\pi_k^2(t)\mathrm{d}\lambda_j(t) \\ &= \sum_j \beta_0^{(j)} \boldsymbol{e}_1^T \boldsymbol{J}^{(j)} [\pi_k(\boldsymbol{J}^{(j)})]^2 \boldsymbol{e}_1 \\ &= \sum_j \beta_0^{(j)} \boldsymbol{e}_1^T [\pi_k(\boldsymbol{J}^{(j)})]^T \boldsymbol{J}^{(j)} \pi_k(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1\end{aligned}$$

similarly

$$(\pi_k, \pi_k)_{\mathrm{d}\lambda} = \sum_j \beta_0^{(j)} \boldsymbol{e}_1^T [\pi_k(\boldsymbol{J}^{(j)})]^T \pi_k(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1$$

Algorithm

$$\zeta_k^{(j)} := \pi_k(\boldsymbol{J}^{(j)})\boldsymbol{e}_1, \quad \boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$$

initialization:

$$\zeta_0^{(j)} = \boldsymbol{e}_1, \ \ \zeta_{-1}^{(j)} = 0 \quad (\text{all } j),$$

$$\alpha_0 = \frac{\sum_j \beta_0^{(j)} \boldsymbol{e}_1^T \boldsymbol{J}^{(j)} \boldsymbol{e}_1}{\sum_j \beta_0^{(j)}}, \quad \beta_0 = \sum_j \beta_0^{(j)}$$

continuation (if $n > 1$): for $k = 0, 1, \ldots, n-2$ do

$$\zeta_{k+1}^{(j)} = (\boldsymbol{J}^{(j)} - \alpha_k \boldsymbol{I})\zeta_k^{(j)} - \beta_k \zeta_{k-1}^{(j)} \quad (\text{all } j),$$

$$\alpha_{k+1} = \frac{\sum_j \beta_0^{(j)} \zeta_{k+1}^{(j)T} \boldsymbol{J}^{(j)} \zeta_{k+1}^{(j)}}{\sum_j \beta_0^{(j)} \zeta_{k+1}^{(j)T} \zeta_{k+1}^{(j)}}, \quad \beta_{k+1} = \frac{\sum_j \beta_0^{(j)} \zeta_{k+1}^{(j)T} \zeta_{k+1}^{(j)}}{\sum_j \beta_0^{(j)} \zeta_k^{(j)T} \zeta_k^{(j)}}$$

Matlab

```
ab=r_multidomain_sti(N,abmd)
```

Solution by modified Chebyshev algorithm

modified moments (by n-point Gauss quadrature, $k \le 2n-1$)

$$m_k = \sum_j \int_{c_j}^{d_j} p_k(t)\mathrm{d}\lambda_j(t) = \sum_j \beta_0^{(j)} \boldsymbol{e}_1^T p_k(\boldsymbol{J}^{(j)})\boldsymbol{e}_1$$

Solution by modified Chebyshev algorithm

modified moments (by n-point Gauss quadrature, $k \le 2n-1$)

$$m_k = \sum_j \int_{c_j}^{d_j} p_k(t) \mathrm{d}\lambda_j(t) = \sum_j \beta_0^{(j)} \boldsymbol{e}_1^T p_k(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1$$

Algorithm

$$\boldsymbol{z}_k^{(j)} := p_k(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1, \quad \boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$$

initialization:

$$\boldsymbol{z}_0^{(j)} = \boldsymbol{e}_1, \ \boldsymbol{z}_{-1}^{(j)} = 0 \ (\text{all } j), \ \ m_0 = \sum_j \beta_0^{(j)}$$

continuation: for $k = 0, 1, \ldots, 2n-2$ do

$$\boldsymbol{z}_{k+1}^{(j)} = (\boldsymbol{J}^{(j)} - a_k \boldsymbol{I}) \boldsymbol{z}_k^{(j)} - b_k \boldsymbol{z}_{k-1}^{(j)} \quad (\text{all } j),$$
$$m_{k+1} = \sum_j \beta_0^{(j)} \boldsymbol{z}_{k+1}^{(j)T} \boldsymbol{e}_1$$

$\Longrightarrow$ modified Chebyshev algorithm

Solution by modified Chebyshev algorithm

modified moments (by n-point Gauss quadrature, $k \leq 2n-1$)

$$m_k = \sum_j \int_{c_j}^{d_j} p_k(t) \mathrm{d}\lambda_j(t) = \sum_j \beta_0^{(j)} \boldsymbol{e}_1^T p_k(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1$$

Algorithm

$$\boldsymbol{z}_k^{(j)} := p_k(\boldsymbol{J}^{(j)}) \boldsymbol{e}_1, \quad \boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$$

initialization:

$$\boldsymbol{z}_0^{(j)} = \boldsymbol{e}_1,\ \boldsymbol{z}_{-1}^{(j)} = 0 \ (\text{all } j),\ \ m_0 = \sum_j \beta_0^{(j)}$$

continuation: for $k = 0, 1, \ldots, 2n-2$ do

$$\boldsymbol{z}_{k+1}^{(j)} = (\boldsymbol{J}^{(j)} - a_k \boldsymbol{I}) \boldsymbol{z}_k^{(j)} - b_k \boldsymbol{z}_{k-1}^{(j)} \quad (\text{all } j),$$
$$m_{k+1} = \sum_j \beta_0^{(j)} \boldsymbol{z}_{k+1}^{(j)T} \boldsymbol{e}_1$$

$\Longrightarrow$ modified Chebyshev algorithm

Matlab `ab=r_multidomain_cheb(N,abmd,abmm)`

Quadrature estimates of matrix functionals

Problem Given $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ postive definite, and f sufficiently smooth, find lower and upper bounds for the quadratic form

$$\boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u}, \quad \boldsymbol{u} \in \mathbb{R}^N, \ \|\boldsymbol{u}\| = 1$$

Quadrature estimates of matrix functionals

Problem Given $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ postive definite, and f sufficiently smooth, find lower and upper bounds for the quadratic form

$$\boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u}, \quad \boldsymbol{u} \in \mathbb{R}^N,\ \|\boldsymbol{u}\| = 1$$

connection with quadrature (spectral resolution of $\boldsymbol{A}$)

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda},\ \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N),\ \boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]$$

$$\boldsymbol{u} = \textstyle\sum_{k=1}^N \rho_k \boldsymbol{v}_k = \boldsymbol{V}\boldsymbol{\rho},\ \boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_N]^T$$

$$\Longrightarrow\ \boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u} = \boldsymbol{\rho}^T\boldsymbol{V}^T\boldsymbol{V} f(\boldsymbol{\Lambda})\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\rho} = \boldsymbol{\rho}^T f(\boldsymbol{\Lambda})\boldsymbol{\rho},$$

$$= \textstyle\sum_{k=1}^N \rho_k^2 f(\lambda_k) =: \int_{\mathbb{R}_+} f(t)\mathrm{d}\rho_N(t)$$

Quadrature estimates of matrix functionals

Problem Given $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ postive definite, and f sufficiently smooth, find lower and upper bounds for the quadratic form

$$\boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u}, \quad \boldsymbol{u} \in \mathbb{R}^N,\ \|\boldsymbol{u}\| = 1$$

connection with quadrature (spectral resolution of $\boldsymbol{A}$)

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda},\ \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N),\ \boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]$$

$$\boldsymbol{u} = \textstyle\sum_{k=1}^N \rho_k \boldsymbol{v}_k = \boldsymbol{V}\boldsymbol{\rho},\ \boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_N]^T$$

$$\Longrightarrow\ \boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u} = \boldsymbol{\rho}^T\boldsymbol{V}^T\boldsymbol{V} f(\boldsymbol{\Lambda})\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\rho} = \boldsymbol{\rho}^T f(\boldsymbol{\Lambda})\boldsymbol{\rho},$$

$$= \textstyle\sum_{k=1}^N \rho_k^2 f(\lambda_k) =: \int_{\mathbb{R}_+} f(t)\mathrm{d}\rho_N(t)$$

Solution (for f with derivatives of constant sign)

- generate $\boldsymbol{J}_N(\mathrm{d}\rho_N)$: Lanczos algorithm
- apply Gauss-type quadrature to

$$\boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u} = \int_{\mathbb{R}_+} f(t)\mathrm{d}\rho_N(t)$$

Lanczos algorithm

$$\boldsymbol{h}_0 = \sum_{k=1}^{N} \rho_k \boldsymbol{v}_k, \quad \|\boldsymbol{h}_0\| = 1 \;\; (\boldsymbol{h}_0 = \boldsymbol{u})$$

initialization:

$$\boldsymbol{h}_0 \text{ prescribed with } \|\boldsymbol{h}_0\| = 1, \qquad \boldsymbol{h}_{-1} = \boldsymbol{0}$$

continuation: for $j = 0, 1, \ldots, N-1$ do

$$\begin{aligned}
\alpha_j &= \boldsymbol{h}_j^T \boldsymbol{A} \boldsymbol{h}_j, \\
\tilde{\boldsymbol{h}}_{j+1} &= (\boldsymbol{A} - \alpha_j \boldsymbol{I})\boldsymbol{h}_j - \gamma_j \boldsymbol{h}_{j-1}, \\
\gamma_{j+1} &= \|\tilde{\boldsymbol{h}}_{j+1}\|, \\
\boldsymbol{h}_{j+1} &= \tilde{\boldsymbol{h}}_{j+1}/\gamma_{j+1}
\end{aligned}$$

Lanczos algorithm

$$\boldsymbol{h}_0 = \sum_{k=1}^{N} \rho_k \boldsymbol{v}_k, \quad \|\boldsymbol{h}_0\| = 1 \;\; (\boldsymbol{h}_0 = \boldsymbol{u})$$

initialization:

$$\boldsymbol{h}_0 \text{ prescribed with } \|\boldsymbol{h}_0\| = 1, \quad \boldsymbol{h}_{-1} = \boldsymbol{0}$$

continuation: for $j = 0, 1, \ldots, N-1$ do

$$\begin{aligned}
\alpha_j &= \boldsymbol{h}_j^T \boldsymbol{A} \boldsymbol{h}_j, \\
\tilde{\boldsymbol{h}}_{j+1} &= (\boldsymbol{A} - \alpha_j \boldsymbol{I})\boldsymbol{h}_j - \gamma_j \boldsymbol{h}_{j-1}, \\
\gamma_{j+1} &= \|\tilde{\boldsymbol{h}}_{j+1}\|, \\
\boldsymbol{h}_{j+1} &= \tilde{\boldsymbol{h}}_{j+1}/\gamma_{j+1}
\end{aligned}$$

Theorem

$$\boldsymbol{J}_N(\mathrm{d}\rho_N) = \begin{bmatrix} \alpha_0 & \gamma_1 & & & \boldsymbol{0} \\ \gamma_1 & \alpha_1 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{N-2} & \alpha_{N-2} & \gamma_{N-1} \\ \boldsymbol{0} & & & \gamma_{N-1} & \alpha_{N-1} \end{bmatrix}$$

Example

error bounds for linear algebraic systems

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A} \text{ symmetric, positive definite}$$

if $\boldsymbol{x}^* \approx \boldsymbol{x} := \boldsymbol{A}^{-1}\boldsymbol{b}$ and $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*$ then

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 = \boldsymbol{r}^T\boldsymbol{A}^{-2}\boldsymbol{r}$$

hence

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 = \|\boldsymbol{r}\|^2 \cdot \boldsymbol{u}^T f(\boldsymbol{A})\boldsymbol{u}$$

where $\boldsymbol{u} = \boldsymbol{r}/\|\boldsymbol{r}\|$, $f(t) = t^{-2}$

Example

error bounds for linear algebraic systems

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A} \text{ symmetric, positive definite}$$

if $\boldsymbol{x}^* \approx \boldsymbol{x} := \boldsymbol{A}^{-1}\boldsymbol{b}$ and $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*$ then

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 = \boldsymbol{r}^T \boldsymbol{A}^{-2} \boldsymbol{r}$$

hence

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 = \|\boldsymbol{r}\|^2 \cdot \boldsymbol{u}^T f(\boldsymbol{A}) \boldsymbol{u}$$

where $\boldsymbol{u} = \boldsymbol{r}/\|\boldsymbol{r}\|$, $f(t) = t^{-2}$

derivatives of f

$$f^{(2n)}(t) > 0, \;\; f^{(2n+1)}(t) < 0 \quad \text{for } t \in \mathbb{R}_+$$

Example (cont')

$$\|\boldsymbol{x} - \boldsymbol{x}^*\| = \|\boldsymbol{r}\|^2 \int_{\mathbb{R}_+} t^{-2} \mathrm{d}\rho_N(t)$$

assume $\quad \operatorname{supp} \mathrm{d}\rho_N \subseteq [a, b], \quad 0 < a < b$

Example (cont')

$$\|\boldsymbol{x} - \boldsymbol{x}^*\| = \|\boldsymbol{r}\|^2 \int_{\mathbb{R}_+} t^{-2} \mathrm{d}\rho_N(t)$$

assume $\operatorname{supp} \mathrm{d}\rho_N \subseteq [a, b], \quad 0 < a < b$

lower bounds for $\|\boldsymbol{x} - \boldsymbol{x}^*\|^2$

- n-point Gauss applied to $\int_{\mathbb{R}_+} t^{-2} \mathrm{d}\rho_N(t)$
- right-handed $(n+1)$-point Gauss-Radau

Example (cont')

$$\|\boldsymbol{x}-\boldsymbol{x}^*\| = \|\boldsymbol{r}\|^2 \int_{\mathbb{R}_+} t^{-2} \mathrm{d}\rho_N(t)$$

assume $\quad \operatorname{supp} \mathrm{d}\rho_N \subseteq [a,b], \quad 0 < a < b$

lower bounds for $\|\boldsymbol{x}-\boldsymbol{x}^*\|^2$

- n-point Gauss applied to $\int_{\mathbb{R}_+} t^{-2} \mathrm{d}\rho_N(t)$
- right-handed $(n+1)$-point Gauss-Radau

upper bounds for $\|\boldsymbol{x}-\boldsymbol{x}^*\|^2$

- left-handed $(n+1)$-point Gauss-Radau
- $(n+2)$-point Gauss-Lobatto

Part II

Approximation

Least squares approximation

classical least squares problem: given N data points (t_k, f_k), $k = 1, 2, \ldots, N$, and the discrete measure $\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - t_k)$, find $\hat{p}_n \in \mathbb{P}_n$, $n < N$, such that

$$\|\hat{p}_n - f\|^2_{\mathrm{d}\lambda_N} \leq \|p - f\|^2_{\mathrm{d}\lambda_N}, \quad \text{all } p \in \mathbb{P}_n$$

Least squares approximation

classical least squares problem: given N data points (t_k, f_k), $k = 1, 2, \ldots, N$, and the discrete measure $\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - t_k)$, find $\hat{p}_n \in \mathbb{P}_n$, $n < N$, such that

$$\|\hat{p}_n - f\|^2_{\mathrm{d}\lambda_N} \leq \|p - f\|^2_{\mathrm{d}\lambda_N}, \quad \text{all } p \in \mathbb{P}_n$$

solution

$$\hat{p}_n(t) = \sum_{i=0}^{n} \hat{c}_i(f)\pi_i(t; \mathrm{d}\lambda_N), \quad \hat{c}_i(f) = \frac{(f, \pi_i)_{\mathrm{d}\lambda_N}}{\|\pi_i\|^2_{\mathrm{d}\lambda_N}}$$

π_i not necessarily monic: $\pi_i(t) = d_i t^i + \cdots$

Least squares approximation

classical least squares problem: given N data points (t_k, f_k), $k = 1, 2, \ldots, N$, and the discrete measure $\mathrm{d}\lambda_N(t) = \sum_{k=1}^{N} w_k \delta(t - t_k)$, find $\hat{p}_n \in \mathbb{P}_n$, $n < N$, such that

$$\|\hat{p}_n - f\|^2_{\mathrm{d}\lambda_N} \leq \|p - f\|^2_{\mathrm{d}\lambda_N}, \quad \text{all } p \in \mathbb{P}_n$$

solution

$$\hat{p}_n(t) = \sum_{i=0}^{n} \hat{c}_i(f)\pi_i(t; \mathrm{d}\lambda_N), \quad \hat{c}_i(f) = \frac{(f, \pi_i)_{\mathrm{d}\lambda_N}}{\|\pi_i\|^2_{\mathrm{d}\lambda_N}}$$

π_i not necessarily monic: $\pi_i(t) = d_i t^i + \cdots$

Matlab

```
[phat,c]=least_squares(n,f,xw,ab,d)
```

Demo #5

L_2 error vs L_∞ error for equally weighted least squares approximation on equally spaced points on $[-1, 1]$

Demo #5

L_2 error vs L_∞ error for equally weighted least squares approximation on equally spaced points on $[-1, 1]$

```
N=10; k=(1:N)'; d=ones(1,N);
xw(k,1)=-1+2*(k-1)/(N-1); xw(:,2)=2/N;
ab=r_hahn(N-1);ab(:,1)=-1+2*ab(:,1)/(N-1);
ab(:,2)=(2/(N-1))^2*ab(:,2); ab(1,2)=2;
[phat,c]=least_squares(N-1,f,xw,ab,d);
```

Demo #5

L_2 error vs L_∞ error for equally weighted least squares approximation on equally spaced points on $[-1, 1]$

```
N=10; k=(1:N)'; d=ones(1,N);
xw(k,1)=-1+2*(k-1)/(N-1); xw(:,2)=2/N;
ab=r_hahn(N-1);ab(:,1)=-1+2*ab(:,1)/(N-1);
ab(:,2)=(2/(N-1))^2*ab(:,2); ab(1,2)=2;
[phat,c]=least_squares(N-1,f,xw,ab,d);
```

Example $\quad f(t) = \ln(2 - t), -1 \le t \le 1$; N=10

n	$\bar{E}_n$	E_n^∞
0	1.88(−01)	6.37(−01)
3	2.96(−03)	3.19(−03)
6	2.07(−05)	7.06(−05)
9	1.74(−16)	3.11(−06)

Constrained least squares approximation

constraints

$$p(s_j) = f_j, \;\; j = 1, 2, \ldots, m; \quad m \leq n$$

interpolation and constraint polynomials

$$p_m(t) = p_m(f; s_1, \ldots, s_m; t), \; \sigma_m(t) = \prod_{j=1}^{m}(t - s_j)$$

Constrained least squares approximation

constraints

$$p(s_j) = f_j, \;\; j = 1, 2, \ldots, m; \quad m \leq n$$

interpolation and constraint polynomials

$$p_m(t) = p_m(f; s_1, \ldots, s_m; t), \; \sigma_m(t) = \prod_{j=1}^{m} (t - s_j)$$

reduction to unconstrained least squares

$$\hat{p}_n(t) = p_m(f; s_1, \ldots, s_m; t) + \sigma_m(t) \hat{q}_{n-m}(t)$$

Constrained least squares approximation

constraints

$$p(s_j) = f_j, \;\; j = 1, 2, \ldots, m; \quad m \le n$$

interpolation and constraint polynomials

$$p_m(t) = p_m(f; s_1, \ldots, s_m; t), \; \sigma_m(t) = \prod_{j=1}^{m}(t - s_j)$$

reduction to unconstrained least squares

$$\hat{p}_n(t) = p_m(f; s_1, \ldots, s_m; t) + \sigma_m(t)\hat{q}_{n-m}(t)$$

where

$$\|f^* - \hat{q}_{n-m}\|^2_{d\lambda_N^*} \le \|f^* - q\|_{d\lambda_N^*}, \quad \text{all } q \in \mathbb{P}_{n-m}$$

$$f^*(t) = [s_1, \ldots, s_m, t]f, \quad d\lambda_N^*(t) = \sigma_m^2(t) d\lambda_N(t)$$

modification algorithm

Demo #6

Example Bessel function $J_0(t)$ for $0 \le t \le j_{0,3}$

constraints (m=3)

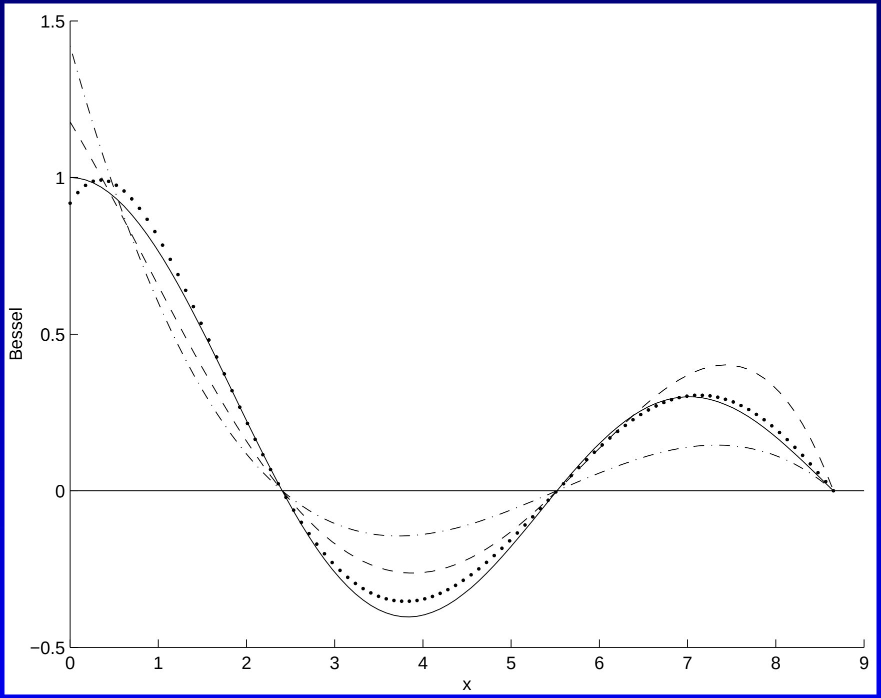
$$s_1 = j_{0,1},\ s_2 = j_{0,2},\ s_3 = j_{0,3};\ \ p(s_j) = 0,\ 1 \le j \le 3$$

Demo #6

Example Bessel function $J_0(t)$ for $0 \leq t \leq j_{0,3}$
constraints (m=3)

$$s_1 = j_{0,1},\ s_2 = j_{0,2},\ s_3 = j_{0,3};\ \ p(s_j) = 0,\ 1 \leq j \leq 3$$

constrained least squares $n = 3, 4, 5$



Demo #7

Example (cont')

additional constraints (m=5)

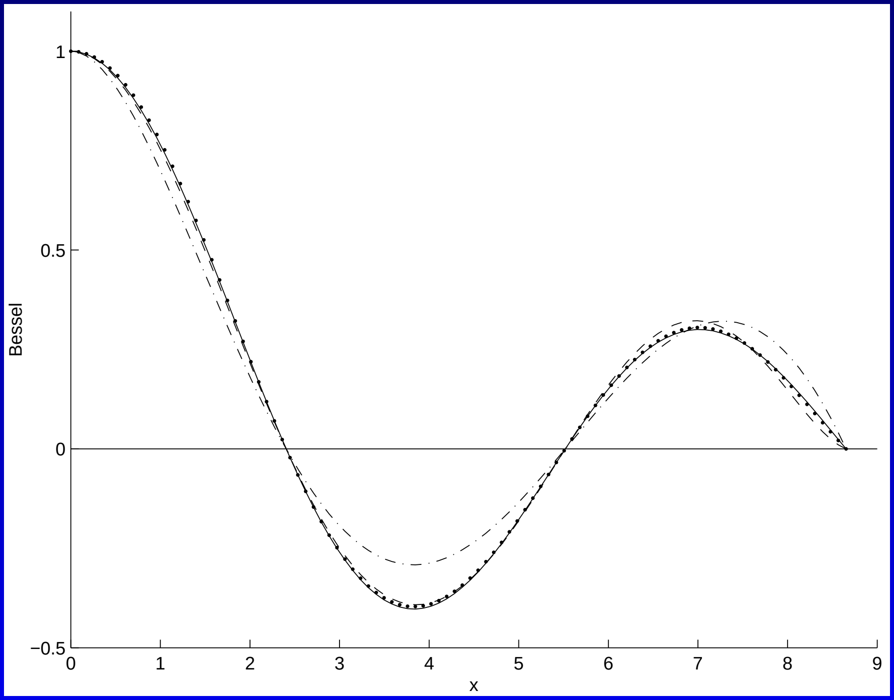
$$p(0) = 1, \quad p'(0) = 0$$

Demo #7

Example (cont')
additional constraints (m=5)

$$p(0) = 1, \quad p'(0) = 0$$

derivative-constrained least squares $n = 5, 6, 7$



L_2 approximation in Sobolev spaces

Problem

$$\text{minimize}: \quad \sum_{\sigma=0}^{s} \sum_{k=1}^{N} w_k^{(\sigma)} [p^{(\sigma)}(t_k) - f_k^{(\sigma)}]^2, \quad p \in \mathbb{P}_n$$

Sobolev inner product and norm

$$(u, v)_S = \sum_{\sigma=0}^{s} \sum_{k=1}^{N} w_k^{(\sigma)} u^{(\sigma)}(t_k) v^{(\sigma)}(t_k), \ \|u\|_S = \sqrt{(u, u)_S}$$

L_2 approximation in Sobolev spaces

Problem

$$\text{minimize}: \quad \sum_{\sigma=0}^{s} \sum_{k=1}^{N} w_k^{(\sigma)} [p^{(\sigma)}(t_k) - f_k^{(\sigma)}]^2, \quad p \in \mathbb{P}_n$$

Sobolev inner product and norm

$$(u, v)_S = \sum_{\sigma=0}^{s} \sum_{k=1}^{N} w_k^{(\sigma)} u^{(\sigma)}(t_k) v^{(\sigma)}(t_k), \ \|u\|_S = \sqrt{(u, u)_S}$$

Solution

$$\hat{p}_n(t) = \sum_{i=0}^{n} \hat{c}_i(f) \pi_i(t), \quad \hat{c}_i(f) = \frac{(f, \pi_i)_S}{\|\pi_i\|_S^2}$$

$\pi_i \ldots$ Sobolev orthogonal polynomials

L_2 approximation in Sobolev spaces

Problem

$$\text{minimize}: \quad \sum_{\sigma=0}^{s} \sum_{k=1}^{N} w_k^{(\sigma)} [p^{(\sigma)}(t_k) - f_k^{(\sigma)}]^2, \quad p \in \mathbb{P}_n$$

Sobolev inner product and norm

$$(u,v)_S = \sum_{\sigma=0}^{s} \sum_{k=1}^{N} w_k^{(\sigma)} u^{(\sigma)}(t_k) v^{(\sigma)}(t_k), \ \|u\|_S = \sqrt{(u,u)_S}$$

Solution

$$\hat{p}_n(t) = \sum_{i=0}^{n} \hat{c}_i(f)\pi_i(t), \quad \hat{c}_i(f) = \frac{(f,\pi_i)_S}{\|\pi_i\|_S^2}$$

π_i ... Sobolev orthogonal polynomials

Matlab

```
[phat,c]=least_squares_sob(n,f,xw,B)
```

Demo #8

Example $\quad f(t) = e^{t^2} \operatorname{erfc} t = \frac{2}{\sqrt{\pi}}\, e^{t^2} \int_t^\infty e^{-u^2} \mathrm{d}u, \quad 0 \le t \le 2$

$$N = 5,\ t_k = 2\tfrac{k-1}{N-1},\ w_k^{(\sigma)} = \tfrac{1}{N},\ \ k = 1, 2, \ldots, N$$

Demo #8

Example $\quad f(t) = e^{t^2} \operatorname{erfc} t = \frac{2}{\sqrt{\pi}} e^{t^2} \int_t^\infty e^{-u^2} \mathrm{d}u, \quad 0 \le t \le 2$

$$N = 5,\ t_k = 2\frac{k-1}{N-1},\ w_k^{(\sigma)} = \frac{1}{N},\ \ k = 1, 2, \ldots, N$$

s	n	$\hat{E}_n$	$E^\infty_{n,0}$	$E^\infty_{n,1}$	$E^\infty_{n,2}$
2	0	1.153(+00)	4.759(−01)	1.128(+00)	2.000(+00)
	2	7.356(−01)	8.812(−02)	2.880(−01)	1.411(+00)
	4	1.196(−01)	1.810(−02)	5.434(−02)	1.980(−01)
	9	2.178(−05)	4.710(−06)	3.011(−05)	3.159(−04)
	14	3.653(−16)	1.130(−09)	1.111(−08)	1.966(−07)
0	0	2.674(−01)	4.759(−01)	1.128(+00)	2.000(+00)
	2	2.245(−02)	3.865(−02)	3.612(−01)	1.590(+00)
	4	1.053(−16)	3.516(−03)	5.160(−02)	4.956(−01)
	9	1.053(−16)	5.409(−03)	8.124(−02)	7.959(−01)
	14	1.053(−16)	5.478(−03)	8.226(−02)	8.057(−01)

Moment-preserving spline approximation

splines of degree 0 (piecewise constant) on $\mathbb{R}_+$

$$s_n(t) = \sum_{\nu=1}^{n} a_\nu H(t_\nu - t), \quad t \in \mathbb{R}_+$$

where $a_\nu \in \mathbb{R}$, $0 < t_1 < t_2 < \cdots < t_n$, and

$$H(u) = \begin{cases} 1 & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Moment-preserving spline approximation

splines of degree 0 (piecewise constant) on $\mathbb{R}_+$

$$s_n(t) = \sum_{\nu=1}^{n} a_\nu H(t_\nu - t), \quad t \in \mathbb{R}_+$$

where $a_\nu \in \mathbb{R}$, $0 < t_1 < t_2 < \cdots < t_n$, and

$$H(u) = \begin{cases} 1 & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Problem Given $f \in C^1(\mathbb{R}_+)$, determine $a_\nu \in \mathbb{R}$ and $0 < t_1 < t_2 < \cdots < t_n$ such that

$$\int_0^\infty s_n(t) t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n-1,$$

where

$$\mu_j = \int_0^\infty f(t) t^j \mathrm{d}t$$

Moment-preserving spline approximation

splines of degree 0 (piecewise constant) on $\mathbb{R}_+$

$$s_n(t) = \sum_{\nu=1}^{n} a_\nu H(t_\nu - t), \quad t \in \mathbb{R}_+$$

where $a_\nu \in \mathbb{R}$, $0 < t_1 < t_2 < \cdots < t_n$, and

$$H(u) = \begin{cases} 1 & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Problem Given $f \in C^1(\mathbb{R}_+)$, determine $a_\nu \in \mathbb{R}$ and $0 < t_1 < t_2 < \cdots < t_n$ such that

$$\int_0^\infty s_n(t) t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n-1,$$

where

$$\mu_j = \int_0^\infty f(t) t^j \mathrm{d}t$$

Solution Gauss quadrature relative to the measure

$$\mathrm{d}\lambda(t) := -t f'(t) \mathrm{d}t \quad \text{on } \mathbb{R}_+$$

Theorem (G., 1984)
If the first $2n$ moments μ_j, $j = 0, 1, \ldots, 2n-1$, exist and $f(t) = o(t^{-2n})$ as $t \to \infty$, then the problem has a unique solution if and only if the measure $\mathrm{d}\lambda(t) = -tf'(t)\mathrm{d}t$ admits a Gaussian quadrature formula

$$\int_0^\infty g(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^n \lambda_\nu^G g(\tau_\nu^G), \quad g \in \mathbb{P}_{2n-1},$$

satisfying $0 < \tau_1^G < \tau_2^G < \cdots < \tau_n^G$. If so, then

$$t_\nu = \tau_\nu^G, \;\; a_\nu = \frac{\lambda_\nu^G}{\tau_\nu^G}, \quad \nu = 1, 2, \ldots, n. \;\square$$

Theorem (G., 1984)
If the first $2n$ moments μ_j, $j = 0, 1, \ldots, 2n-1$, exist and $f(t) = o(t^{-2n})$ as $t \to \infty$, then the problem has a unique solution if and only if the measure $\mathrm{d}\lambda(t) = -tf'(t)\mathrm{d}t$ admits a Gaussian quadrature formula

$$\int_0^\infty g(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^n \lambda_\nu^G g(\tau_\nu^G), \quad g \in \mathbb{P}_{2n-1},$$

satisfying $0 < \tau_1^G < \tau_2^G < \cdots < \tau_n^G$. If so, then

$$t_\nu = \tau_\nu^G, \;\; a_\nu = \frac{\lambda_\nu^G}{\tau_\nu^G}, \quad \nu = 1, 2, \ldots, n. \; \square$$

Remark If $f' < 0$ on $\mathbb{R}_+$, then $\mathrm{d}\lambda(t) \geq 0$.

Proof Integration by parts

$$\int_0^T f(t)t^j \mathrm{d}t = \frac{1}{j+1}\, t^{j+1} f(t)\Big|_0^T - \frac{1}{j+1}\int_0^T f'(t)t^{j+1}\mathrm{d}t$$

letting $T \to \infty$ and recalling $-tf'(t)\mathrm{d}t =: \mathrm{d}\lambda(t)$

$$\mu_j = \frac{1}{j+1}\int_0^\infty t^j \mathrm{d}\lambda(t), \quad j = 0, 1, \ldots, 2n-1$$

Proof Integration by parts

$$\int_0^T f(t)t^j \mathrm{d}t = \tfrac{1}{j+1}\, t^{j+1} f(t)\Big|_0^T - \tfrac{1}{j+1}\,\int_0^T f'(t)t^{j+1}\mathrm{d}t$$

letting $T \to \infty$ and recalling $-tf'(t)\mathrm{d}t =: \mathrm{d}\lambda(t)$

$$\mu_j = \tfrac{1}{j+1}\,\int_0^\infty t^j \mathrm{d}\lambda(t), \quad j = 0, 1, \ldots, 2n-1$$

on the other hand

$$\int_0^\infty s_n(t)t^j\mathrm{d}t = \textstyle\sum_{\nu=1}^n a_\nu \int_0^{t_\nu} t^j \mathrm{d}t = \tfrac{1}{j+1}\,\sum_{\nu=1}^n a_\nu t_\nu^{j+1}$$

Proof Integration by parts

$$\int_0^T f(t)t^j \mathrm{d}t = \frac{1}{j+1}\, t^{j+1} f(t)\Big|_0^T - \frac{1}{j+1}\int_0^T f'(t)t^{j+1}\mathrm{d}t$$

letting $T \to \infty$ and recalling $-tf'(t)\mathrm{d}t =: \mathrm{d}\lambda(t)$

$$\mu_j = \frac{1}{j+1}\int_0^\infty t^j \mathrm{d}\lambda(t), \quad j = 0, 1, \ldots, 2n-1$$

on the other hand

$$\int_0^\infty s_n(t)t^j \mathrm{d}t = \sum_{\nu=1}^n a_\nu \int_0^{t_\nu} t^j \mathrm{d}t = \frac{1}{j+1}\sum_{\nu=1}^n a_\nu t_\nu^{j+1}$$

moment matching

$$\sum_{\nu=1}^n (a_\nu t_\nu) t_\nu^j = \int_0^\infty t^j \mathrm{d}\lambda(t), \quad j = 0, 1, \ldots, 2n-1$$

$\square$

Example Maxwell distribution (Calder & Laframboise, 1986)

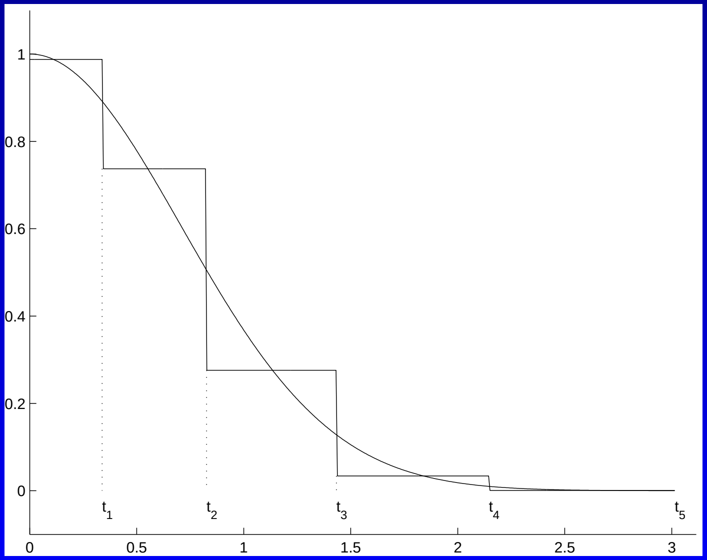
$$f(t) = e^{-t^2}, \;\; \mathrm{d}\lambda(t) = 2t^2 e^{-t^2}\mathrm{d}t \text{ on } \mathbb{R}_+$$

The recurrence coefficients $\alpha_k(\mathrm{d}\lambda)$, $\beta_k(\mathrm{d}\lambda)$, $k \leq n-1$, and hence the Gauss quadrature rule for $\mathrm{d}\lambda$, can be obtained by twice modifying the half-range Hermite measure by a linear factor t (two applications of `chri1`).

Example Maxwell distribution (Calder & Laframboise, 1986)

$$f(t) = e^{-t^2}, \;\; d\lambda(t) = 2t^2 e^{-t^2} dt \text{ on } \mathbb{R}_+$$

The recurrence coeffi cients $\alpha_k(d\lambda)$, $\beta_k(d\lambda)$, $k \le n-1$, and hence the Gauss quadrature rule for $d\lambda$, can be obtained by twice modifying the half-range Hermite measure by a linear factor t (two applications of `chri1`).



spline function of degree $m > 0$

$$s_{n,m}(t) = \sum_{\nu=1}^{n} a_\nu (t_\nu - t)_+^m, \quad t \in \mathbb{R}_+$$

where

$$u_+ = uH(u) = \begin{cases} u & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

spline function of degree $m > 0$

$$s_{n,m}(t) = \sum_{\nu=1}^{n} a_\nu (t_\nu - t)_+^m, \quad t \in \mathbb{R}_+$$

where

$$u_+ = uH(u) = \begin{cases} u & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Problem Given $f \in C^{m+1}(\mathbb{R}_+)$ and $\mu_j = \int_{\mathbb{R}_+} f(t) t^j \mathrm{d}t$, $j \leq 2n-1$, determine $a_\nu \in \mathbb{R}$ and $0 < t_1 < \cdots < t_n$ such that

$$\int_{\mathbb{R}_+} s_{n,m}(t) t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n-1$$

spline function of degree $m > 0$

$$s_{n,m}(t) = \sum_{\nu=1}^{n} a_\nu (t_\nu - t)_+^m, \quad t \in \mathbb{R}_+$$

where

$$u_+ = uH(u) = \begin{cases} u & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Problem Given $f \in C^{m+1}(\mathbb{R}_+)$ and $\mu_j = \int_{\mathbb{R}_+} f(t)t^j \mathrm{d}t$, $j \leq 2n-1$, determine $a_\nu \in \mathbb{R}$ and $0 < t_1 < \cdots < t_n$ such that

$$\int_{\mathbb{R}_+} s_{n,m}(t)t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n-1$$

Solution Gauss quadrature relative to

$$\mathrm{d}\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!} t^{m+1} f^{(m+1)}(t)\mathrm{d}t \ \text{ on } \mathbb{R}_+$$

Remark If f is completely monotonic on $\mathbb{R}_+$, then $\mathrm{d}\lambda^{[m]} \geq 0$

Example Maxwell distribution

$$\mathrm{d}\lambda(t) = \frac{1}{m!}\, t^{m+1} H_{m+1}(t) e^{-t^2} \mathrm{d}t \quad \text{on } \mathbb{R}_+$$

Example Maxwell distribution

$$\mathrm{d}\lambda(t) = \frac{1}{m!}\, t^{m+1} H_{m+1}(t) e^{-t^2} \mathrm{d}t \quad \text{on } \mathbb{R}_+$$

existence and accuracy of spline approximant

n	$m=1$	$m=2$	$m=3$	n	$m=1$	$m=2$	$m=3$
1	6.9(−2)	1.8(−1)	2.6(−1)	11	—	1.1(−3)	1.1(−4)
2	8.2(−2)	—	2.3(−1)	12	—	—	*
3	—	1.1(−2)	2.5(−3)	13	7.8(−3)	6.7(−4)	*
4	3.5(−2)	6.7(−3)	2.2(−3)	14	8.3(−3)	5.6(−4)	8.1(−5)
5	2.6(−2)	—	1.6(−3)	15	7.7(−3)	—	7.1(−5)
6	2.2(−2)	3.1(−3)	*	16	—	4.9(−4)	7.8(−5)
7	—	2.4(−3)	*	17	—	3.8(−4)	3.8(−5)
8	1.4(−2)	—	3.4(−4)	18	5.5(−3)	3.8(−4)	*
9	1.1(−2)	1.7(−3)	2.5(−4)	19	5.3(−3)	—	*
10	9.0(−3)	1.1(−3)	—	20	5.4(−3)	3.1(−4)	*

approximation on the interval $[0, 1]$

spline function of degree $m \geq 0$

$$s_{n,m}(t) = p(t) + \sum_{\nu=1}^{n} a_\nu (t_\nu - t)_+^m, \;\; p \in \mathbb{P}_m, \quad 0 \leq t \leq 1$$

where $a_\nu \in \mathbb{R}$, $0 < t_1 < t_2 < \cdots < t_n < 1$

approximation on the interval $[0,1]$

spline function of degree $m \geq 0$

$$s_{n,m}(t) = p(t) + \sum_{\nu=1}^{n} a_\nu (t_\nu - t)_+^m, \;\; p \in \mathbb{P}_m, \quad 0 \leq t \leq 1$$

where $a_\nu \in \mathbb{R}$, $0 < t_1 < t_2 < \cdots < t_n < 1$

Problem I Find $s_{n,m}$ such that

$$\int_0^1 s_{n,m}(t) t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n+m$$

approximation on the interval $[0,1]$

spline function of degree $m \geq 0$

$$s_{n,m}(t) = p(t) + \sum_{\nu=1}^{n} a_\nu (t_\nu - t)_+^m, \;\; p \in \mathbb{P}_m, \quad 0 \leq t \leq 1$$

where $a_\nu \in \mathbb{R}$, $0 < t_1 < t_2 < \cdots < t_n < 1$

Problem I   Find $s_{n,m}$ such that

$$\int_0^1 s_{n,m}(t) t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n+m$$

Problem II   Find $s_{n,m}$ such that

$$\int_0^1 s_{n,m}(t) t^j \mathrm{d}t = \mu_j, \quad j = 0, 1, \ldots, 2n-1$$

and

$$s_{n,m}^{(\mu)}(1) = f^{(\mu)}(1), \quad \mu = 0, 1, \ldots, m$$

Solution of Problem I (Frontini, G., and Milovanović, 1987)

generalized Gauss-Lobatto quadrature

$$\mathrm{d}\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!}\, f^{(m+1)}(t)\mathrm{d}t \quad \text{on } [0,1]$$

Solution of Problem I (Frontini, G., and Milovanović, 1987)

generalized Gauss-Lobatto quadrature

$$\mathrm{d}\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!} f^{(m+1)}(t)\mathrm{d}t \quad \text{on } [0,1]$$

$$\int_0^1 g(t)\mathrm{d}\lambda^{[m]}(t) = \sum_{\mu=0}^{m}[\lambda_0^{(\mu)} g^{(\mu)}(0) + (-1)^\mu \lambda_{n+1}^{(\mu)} g^{(\mu)}(1)] + \sum_{\nu=1}^{n} \lambda_\nu^L g(\tau_\nu^L), \quad g \in \mathbb{P}_{2n+2m+1}$$

if this exists with $0 < \tau_1^L < \cdots < \tau_n^L < 1$, then

$$t_\nu = \tau_\nu^L, \;\; a_\nu = \lambda_\nu^L, \quad \nu = 1, 2, \ldots .n$$

and p is uniquely determined by

$$p^{(\mu)}(1) = f^{(\mu)}(1) + (-1)^m m! \lambda_{n+1}^{(m-\mu)}, \quad \mu = 0, 1, \ldots, m$$

Solution of Problem I (Frontini, G., and Milovanović, 1987)
generalized Gauss-Lobatto quadrature

$$\mathrm{d}\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!}\, f^{(m+1)}(t)\mathrm{d}t \quad \text{on } [0,1]$$

$$\int_0^1 g(t)\mathrm{d}\lambda^{[m]}(t) = \sum_{\mu=0}^{m}[\lambda_0^{(\mu)} g^{(\mu)}(0) + (-1)^\mu \lambda_{n+1}^{(\mu)} g^{(\mu)}(1)] + \sum_{\nu=1}^{n} \lambda_\nu^L g(\tau_\nu^L), \quad g \in \mathbb{P}_{2n+2m+1}$$

if this exists with $0 < \tau_1^L < \cdots < \tau_n^L < 1$, then

$$t_\nu = \tau_\nu^L,\ \ a_\nu = \lambda_\nu^L, \quad \nu = 1, 2, \ldots . n$$

and p is uniquely determined by

$$p^{(\mu)}(1) = f^{(\mu)}(1) + (-1)^m m! \lambda_{n+1}^{(m-\mu)}, \quad \mu = 0, 1, \ldots, m$$

complete monotonicity of f implies $\mathrm{d}\lambda \geq 0$

Solution of Problem II (Frontini, G., and Milovanović, 1987) generalized Gauss-Radau quadrature

$$\mathrm{d}\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!} f^{(m+1)}(t)\mathrm{d}t \quad \text{on } [0,1]$$

Solution of Problem II (Frontini, G., and Milovanović, 1987)
generalized Gauss-Radau quadrature

$$d\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!}\, f^{(m+1)}(t)\mathrm{d}t \quad \text{on } [0,1]$$

$$\int_0^1 g(t)d\lambda^{[m]}(t) = \sum_{\mu=0}^{m} \lambda_0^{(\mu)} g^{(\mu)}(0) + \sum_{\nu=1}^{n} \lambda_\nu^R g(\tau_\nu^R), \;\; g \in \mathbb{P}_{2n}$$

if this exists with $0 < \tau_1^R < \cdots < \tau_n^R < 1$, then

$$t_\nu = \tau_\nu^R, \;\; a_\nu = \lambda_\nu^R, \quad \nu = 1, 2, \ldots .n$$

and (trivially)

$$p(t) = \sum_{\mu=0}^{m} \frac{f^{(\mu)}(1)}{\mu!}\,(t-1)^\mu$$

Solution of Problem II (Frontini, G., and Milovanović, 1987)

generalized Gauss-Radau quadrature

$$d\lambda^{[m]}(t) = \frac{(-1)^{m+1}}{m!}\, f^{(m+1)}(t)dt \quad \text{on } [0,1]$$

$$\int_0^1 g(t) d\lambda^{[m]}(t) = \sum_{\mu=0}^{m} \lambda_0^{(\mu)} g^{(\mu)}(0) + \sum_{\nu=1}^{n} \lambda_\nu^R g(\tau_\nu^R), \;\; g \in \mathbb{P}_{2n-}$$

if this exists with $0 < \tau_1^R < \cdots < \tau_n^R < 1$, then

$$t_\nu = \tau_\nu^R, \;\; a_\nu = \lambda_\nu^R, \quad \nu = 1, 2, \ldots .n$$

and (trivially)

$$p(t) = \sum_{\mu=0}^{m} \frac{f^{(\mu)}(1)}{\mu!} (t-1)^\mu$$

complete monotonicity of f implies $d\lambda \geq 0$

Slowly convergent series

series generated by a Laplace transform

$$S = \sum_{k=1}^{\infty} a_k, \;\; a_k = (\mathcal{L}\, f)(k), \quad k = 1, 2, 3, \ldots$$

where

$$(\mathcal{L}\, f)(s) = \int_0^{\infty} e^{-st} f(t)\mathrm{d}t$$

Slowly convergent series

series generated by a Laplace transform

$$S = \sum_{k=1}^{\infty} a_k, \;\; a_k = (\mathcal{L}\, f)(k), \quad k = 1, 2, 3, \ldots$$

where

$$(\mathcal{L}\, f)(s) = \int_0^\infty e^{-st} f(t) \mathrm{d}t$$

reduction to a quadrature problem

$$\begin{aligned} S &= \sum_{k=1}^{\infty} \int_0^\infty e^{-kt} f(t) \mathrm{d}t \\ &= \int_0^\infty \sum_{k=1}^{\infty} e^{-(k-1)t} \cdot e^{-t} f(t) \mathrm{d}t \\ &= \int_0^\infty \frac{1}{1-e^{-t}}\, e^{-t} f(t) \mathrm{d}t \\ &= \int_0^\infty \frac{t}{1-e^{-t}}\, \frac{f(t)}{t}\, e^{-t} \mathrm{d}t \end{aligned}$$

$$S = \int_0^\infty \frac{t}{1-e^{-t}} \frac{f(t)}{t} e^{-t} \mathrm{d}t$$

quadrature methods

- Gauss-Laguerre $\mathrm{d}\lambda(t) = e^{-t}\mathrm{d}t$ on $\mathbb{R}_+$
- rational/polynomial Gauss-Laguerre
- Gauss-Einstein $\mathrm{d}\lambda(t) = \frac{t}{e^t-1}\,\mathrm{d}t$

$$S = \int_0^\infty \frac{t}{1-e^{-t}} \frac{f(t)}{t} e^{-t} \mathrm{d}t$$

quadrature methods

- Gauss-Laguerre $\mathrm{d}\lambda(t) = e^{-t}\mathrm{d}t$ on $\mathbb{R}_+$
- rational/polynomial Gauss-Laguerre
- Gauss-Einstein $\mathrm{d}\lambda(t) = \frac{t}{e^t-1}\,\mathrm{d}t$

Example Theodorus constant (P.J. Davis, 1993)

$$S = \sum_{k=1}^\infty \frac{1}{k^{3/2}+k^{1/2}} = 1.860025\ldots$$

here

$$\frac{1}{s^{3/2}+s^{1/2}} = s^{-1/2}\frac{1}{s+1} = \left(\mathcal{L}\frac{1}{\sqrt{\pi t}} \underset{\substack{\uparrow \\ \text{convolution}}}{*} e^{-t}\right)(s)$$

$$f(t) = \frac{2}{\sqrt{\pi}} F(\sqrt{t}), \quad F(x) := e^{-x^2}\int_0^x e^{t^2}\mathrm{d}t$$

Demo #9 Theodorus constant

$$S = \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{t}{1-e^{-t}} \frac{F(\sqrt{t})}{\sqrt{t}} t^{-1/2} e^{-t} \mathrm{d}t$$

$$= \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{F(\sqrt{t})}{\sqrt{t}} t^{-1/2} \frac{t}{e^t - 1} \, \mathrm{d}t$$

Demo #9 Theodorus constant

$$S = \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{t}{1-e^{-t}} \frac{F(\sqrt{t})}{\sqrt{t}} t^{-1/2} e^{-t} \mathrm{d}t$$

$$= \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{F(\sqrt{t})}{\sqrt{t}} t^{-1/2} \frac{t}{e^t - 1} \mathrm{d}t$$

n	Gauss-Laguerre	rational Gauss-Laguerre	Gauss-Einstein
1	9.6799(−03)	1.5635(−02)	1.3610(−01)
4	5.5952(−06)	1.1893(−08)	2.1735(−04)
7	4.0004(−08)	5.9689(−16)	3.3459(−07)
10	5.9256(−10)		5.0254(−10)
15	8.2683(−12)		9.4308(−15)
20	8.9175(−14)		4.7751(−16)
	timing: 10.8	timing: 8.78	timing: 10.4

Example Hardy-Littlewood function

$$H(x) = \sum_{k=1}^{\infty} \frac{1}{k} \sin \frac{x}{k}, \quad x > 0$$

inverse Laplace transform of $\frac{1}{s} \sin \frac{x}{s}$

$$f(t;x) = \frac{1}{2i} \left[I_0(2\sqrt{ixt}) - I_0(2\sqrt{-ixt}) \right]$$

integral representation for H

$$H(x) = \int_0^\infty \frac{t}{1-e^{-t}} \frac{f(t;x)}{t} e^{-t} \mathrm{d}t = \int_0^\infty \frac{f(t;x)}{t} \frac{t}{e^t-1} \mathrm{d}t$$

Example Hardy-Littlewood function

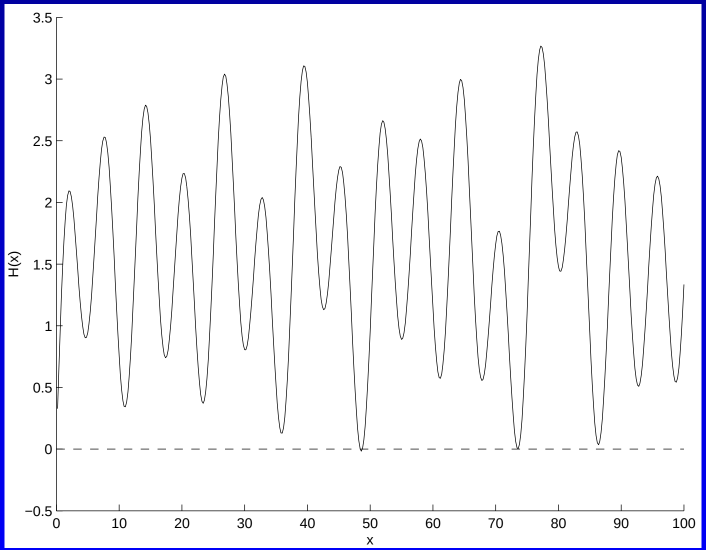
$$H(x) = \sum_{k=1}^{\infty} \frac{1}{k}\,\sin\frac{x}{k}, \quad x > 0$$

inverse Laplace transform of $\frac{1}{s}\,\sin\frac{x}{s}$

$$f(t;x) = \frac{1}{2i}\left[I_0(2\sqrt{ixt}) - I_0(2\sqrt{-ixt})\right]$$

integral representation for H

$$H(x) = \int_0^{\infty} \frac{t}{1-e^{-t}}\,\frac{f(t;x)}{t}\,e^{-t}\mathrm{d}t = \int_0^{\infty} \frac{f(t;x)}{t}\,\frac{t}{e^{t}-1}\mathrm{d}t$$



"alternating" series generated by a Laplace transform

$$S = \sum_{k=1}^{\infty} a_k, \quad a_k = (-1)^{k-1}(\mathcal{L}\, f)(k), \qquad k = 1, 2, 3, \ldots$$

$$S = \int_0^{\infty} \frac{1}{1+e^{-t}}\, f(t) e^{-t} \mathrm{d}t = \int_0^{\infty} f(t) \frac{1}{e^t+1}\, \mathrm{d}t$$

Gauss-Laguerre, rat/pol Gauss-Laguerre, or Gauss-Fermi "alternating" series generated by a Laplace transform

$$S = \sum_{k=1}^{\infty} a_k, \quad a_k = (-1)^{k-1}(\mathcal{L} f)(k), \qquad k = 1, 2, 3, \ldots$$

$$S = \int_0^\infty \frac{1}{1+e^{-t}} f(t) e^{-t} \mathrm{d}t = \int_0^\infty f(t) \frac{1}{e^t+1} \mathrm{d}t$$

Gauss-Laguerre, rat/pol Gauss-Laguerre, or Gauss-Fermi

Example $\sum_{k=1}^{\infty}(-1)^{k-1}k^{-1}e^{-1/k}$, $f(t) = J_0(2\sqrt{t})$

n	Gauss-Laguerre	rational Gauss-Laguerre	Gauss-Fermi
1	1.6961(−01)	1.0310(−01)	5.6994(−01)
4	4.4754(−03)	4.6605(−05)	9.6454(−07)
7	1.7468(−04)	1.8274(−09)	9.1529(−15)
10	3.7891(−06)	1.5729(−13)	2.8163(−16)
15	2.6569(−07)	1.5490(−15)	
20	8.6155(−09)		
40	1.8066(−13)		
	timing: 12.7	timing: 19.5	timing: 4.95

series generated by the derivative of a Laplace transform

$$S = \sum_{k=1}^{\infty} a_k, \quad a_k = -\frac{\mathrm{d}}{\mathrm{d}s}(\mathcal{L}f)(s)\Big|_{s=k}, \qquad k = 1, 2, 3, \ldots$$

$$S = \int_0^{\infty} \frac{t}{1-e^{-t}}\, f(t) e^{-t} \mathrm{d}t = \int_0^{\infty} f(t) \frac{t}{e^t - 1}\, \mathrm{d}t$$

Gauss-Laguerre, rat/pol Gauss-Laguerre, or Gauss-Einstein series generated by the derivative of a Laplace transform

$$S = \sum_{k=1}^{\infty} a_k, \quad a_k = -\frac{\mathrm{d}}{\mathrm{d}s}(\mathcal{L}f)(s)\Big|_{s=k}, \qquad k = 1, 2, 3, \ldots$$

$$S = \int_0^{\infty} \frac{t}{1-e^{-t}}\, f(t) e^{-t} \mathrm{d}t = \int_0^{\infty} f(t) \frac{t}{e^t - 1}\, \mathrm{d}t$$

Gauss-Laguerre, rat/pol Gauss-Laguerre, or Gauss-Einstein

Example $\sum_{k=1}^{\infty} (\frac{3}{2} + 1) k^{-2} (k+1)^{-3/2}$, $f(t) = \frac{\operatorname{erf}\sqrt{t}}{\sqrt{t}} \cdot t^{1/2}$

n	Gauss-Laguerre	rational Gauss-Laguerre	Gauss-Einstein
1	4.0125(−03)	5.1071(−02)	8.1715(−02)
4	1.5108(−05)	4.5309(−08)	1.6872(−04)
7	4.6576(−08)	1.3226(−13)	3.1571(−07)
10	3.0433(−09)	1.2087(−15)	5.4661(−10)
15	4.3126(−11)		1.2605(−14)
20	7.6664(−14)		
30	3.4533(−16)		
	timing: 6.50	timing: 10.8	timing: 1.58

Slowly convergent series occurring in plate contact problems

$$R_p(z) = \sum_{k=0}^{\infty} \frac{z^{2k+1}}{(2k+1)^p}, \quad z \in \mathbb{C},\ |z| \le 1,\ p = 2 \text{ or } 3$$

Slowly convergent series occurring in plate contact problems

$$R_p(z) = \sum_{k=0}^{\infty} \frac{z^{2k+1}}{(2k+1)^p}, \qquad z \in \mathbb{C},\ |z| \le 1,\ p = 2 \text{ or } 3$$

let

$$\frac{1}{(k+\frac{1}{2})^p} = (\mathcal{L}f)(k), \qquad f(t) = \frac{1}{(p-1)!}\, t^{p-1} e^{-t/2}$$

then

$$\begin{aligned}
R_p(z) &= \frac{z}{2^p} \sum_{k=0}^{\infty} \frac{z^{2k}}{(k+\frac{1}{2})^p} \\
&= \frac{z}{2^p} \sum_{k=0}^{\infty} z^{2k} \int_0^{\infty} e^{-kt} \cdot \frac{t^{p-1} e^{-t/2}}{(p-1)!}\, \mathrm{d}t \\
&= \frac{z}{2^p (p-1)!} \int_0^{\infty} \sum_{k=0}^{\infty} (z^2 e^{-t})^k \cdot t^{p-1} e^{-t/2} \mathrm{d}t \\
&= \frac{z}{2^p (p-1)!} \int_0^{\infty} \frac{1}{1 - z^2 e^{-t}}\, t^{p-1} e^{-t/2} \mathrm{d}t
\end{aligned}$$

$$R_p(z) = \frac{z}{2^p(p-1)!} \int_0^\infty \frac{t^{p-1}e^{t/2}}{e^t - z^2}\,\mathrm{d}t, \qquad z^{-2} \in \mathbb{C}\backslash[0,1]$$

change of variables $e^{-t} \mapsto t$

$$R_p(z) = \frac{1}{2^p(p-1)!z} \int_0^1 \frac{t^{-1/2}[\ln(1/t)]^{p-1}}{z^{-2} - t}\,\mathrm{d}t$$

$$R_p(z) = \frac{z}{2^p(p-1)!}\int_0^\infty \frac{t^{p-1}e^{t/2}}{e^t - z^2}\,\mathrm{d}t, \qquad z^{-2} \in \mathbb{C}\backslash[0,1]$$

change of variables $e^{-t} \mapsto t$

$$R_p(z) = \frac{1}{2^p(p-1)!z}\int_0^1 \frac{t^{-1/2}[\ln(1/t)]^{p-1}}{z^{-2} - t}\,\mathrm{d}t$$

Cauchy integral of $\mathrm{d}\lambda^{[p]}(t) = t^{-1/2}[\ln(1/t)]^{p-1}\mathrm{d}t$

continued fraction algorithm in combination with modified Chebyshev algorithm to generate recurrence coefficients of $\mathrm{d}\lambda^{[p]}$

Example

$R_p(x)$, $p = 2$ and 3, $x = .8, .9, .95, .99, .999$ and 1.0

x	$R_2(x)$	$R_3(x)$
.8	0.87728809392147	0.82248858052014
.9	1.02593895111111	0.93414857586540
.95	1.11409957792905	0.99191543992243
.99	1.20207566477686	1.03957223187364
.999	1.22939819733	1.05056774973
1.000	1.233625	1.051795

for $x \geq .999$ full accuracy cannot be achieved with `numax`=100 recurrence coefficients

Example

$R_p(e^{i\alpha})$, $p = 2$ and 3, $\alpha = \omega\pi/2$,
$\omega = .2, .1, .05, .01, .001$ and 0.0

p	ω	$\mathrm{Re}(R_p(z))$	$\mathrm{Im}(R_p(z))$
2	.2	0.98696044010894	0.44740227008596
3		0.96915102126252	0.34882061265337
2	0.1	1.11033049512255	0.27830297928558
3		1.02685555765937	0.18409976778928
2	0.05	1.17201552262936	0.16639152396897
3		1.04449441539672	0.09447224926029
2	0.01	1.22136354463481	0.04592009281744
3		1.05140829197388	0.01928202831056
2	0.001	1.232466849	0.006400460
3		1.051794454	0.001936923
2	0.000	1.2336	0.0000
3		1.0518	0.0000

Series involving ratios of hyperbolic cosines

$$T_p(x;b) = \sum_{k=0}^{\infty} \frac{1}{(2k+1)^p} \frac{\cosh(2k+1)k}{\cosh(2k+1)b}, \quad 0 \le x \le b,\ b > 0,\ p = 2, 3$$

Series involving ratios of hyperbolic cosines

$$T_p(x;b) = \sum_{k=0}^{\infty} \frac{1}{(2k+1)^p} \frac{\cosh(2k+1)k}{\cosh(2k+1)b}, \quad 0 \leq x \leq b, \; b > 0, \; p = 2, 3$$

same Laplace transform technique, but after expanding

$$\frac{\cosh(2k+1)x}{\cosh(2k+1)b}$$
$$= \sum_{n=0}^{\infty} (-1)^n \left\{ e^{-(2k+1)[(2n+1)b-x]} + e^{-(2k+1)[(2n+1)b+x]} \right\}$$

Series involving ratios of hyperbolic cosines

$$T_p(x;b) = \sum_{k=0}^{\infty} \frac{1}{(2k+1)^p} \frac{\cosh(2k+1)k}{\cosh(2k+1)b}, \quad 0 \le x \le b,\ b > 0,\ p = 2, 3$$

same Laplace transform technique, but after expanding

$$\frac{\cosh(2k+1)x}{\cosh(2k+1)b}$$
$$= \sum_{n=0}^{\infty} (-1)^n \left\{ e^{-(2k+1)[(2n+1)b-x]} + e^{-(2k+1)[(2n+1)b+x]} \right\}$$

result

$$T_p(x,b) = \frac{1}{2^p(p-1)!} \sum_{n=0}^{\infty} (-1)^n e^{(2n+1)b} [\varphi_n(-x) + \varphi_n(x)]$$

where

$$\varphi_n(s) = e^s \int_0^1 \frac{\mathrm{d}\lambda^{[p]}(t)}{e^{2[(2n+1)b+s]} - t}, \qquad -b \le s \le b,$$

continued fraction algorithm for $\mathrm{d}\lambda^{[p]}$