Orthogonal Polynomials (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
- Modification algorithms

Orthogonal Polynomials

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

Sobolev Orthogonal Polynomials

Orthogonal Polynomials

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

Sobolev Orthogonal Polynomials

- Moment-based algorithm

Orthogonal Polynomials

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

Sobolev Orthogonal Polynomials

- Moment-based algorithm
- Discretization algorithm

Orthogonal Polynomials

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

Sobolev Orthogonal Polynomials

- Moment-based algorithm
- Discretization algorithm
- Zeros

Reference W. Gautschi, "Orthogonal Polynomials: Computation and Approximation", Clarendon Press, Oxford, 2004

`http://www.cs.purdue.edu/homes/wxg/papers.html`

click on Madrid.ps

also in: Lecture Notes in Mathematics, 2007(?)

`http://www.cs.purdue.edu/archives/2002/wxg/codes`

Background

inner product

$$(p,q)_{d\lambda} = \int_{\mathbb{R}} p(t)q(t)\,d\lambda(t), \quad d\lambda \geq 0$$

orthogonal polynomials

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

three-term recurrence relation

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

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

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

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

"Classical" Weight Functions

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

name	$w(t)$	on
Jacobi	$(1-t)^{\alpha}(1+t)^{\beta}$	$[-1,1]$
Laguerre	$t^{\alpha}e^{-t}$	$[0,\infty]$
Hermite	$\lvert t\rvert^{2\alpha}e^{-t^2}$	$[-\infty,\infty]$

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 moments

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

Modified Chebyshev Algorithm

modified moments

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

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 moments

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

modified moment map

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

algorithm: Sack et al., 1971; Wheeler, 1974

conditioning: G.,1968; 1982

Matlab

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

Demo #2

elliptic orthogonal polynomials

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

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

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

Demo #2

elliptic orthogonal polynomials

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

Chebyshev moments

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

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

Discrete Measure

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

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

Algorithm #1: Stieltjes, 1884

Discrete Measure

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

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

Algorithm #1: Stieltjes, 1884

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

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

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

Algorithm #1: Stieltjes, 1884

Darboux's formulae

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

plus recurrence relation (R)

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

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)
```

$n \le N$

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

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

example

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

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

basic idea

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

example

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

discretization

$$(p,q) =$$

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

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

in general

discretize $\mathrm{d}\lambda$ on

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

using

$$\left.\begin{array}{l}\text{tailor-made quadratures}\\[2ex]\text{general-purpose quadratures}\end{array}\right\}\ \text{on each } [a_j, b_j]$$

Matlab

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

structure (global variables)

$$mc,\ mp,\ iq$$

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

example $w(t) = t^\alpha K_0(t)$ on $[0, \infty]$, $\alpha > -1$

example $w(t) = t^{\alpha} K_0(t)$ on $[0, \infty]$, $\alpha > -1$

$$K_0(t) = \begin{cases} R(t) + I_0(t) \ln(1/t) & \text{if } 0 < t \leq 1, \\ t^{-1/2} e^{-t} S(t) & \text{if } 1 \leq t < \infty, \end{cases}$$

example $\quad w(t) = t^\alpha K_0(t)$ on $[0,\infty]$, $\quad \alpha > -1$

$$K_0(t) = \begin{cases} R(t) + I_0(t)\ln(1/t) & \text{if } 0 < t \le 1, \\ t^{-1/2} e^{-t} S(t) & \text{if } 1 \le t < \infty, \end{cases}$$

$$\int_0^\infty f(t)w(t)\mathrm{d}t$$
$$= \int_0^1 [R(t)f(t)]t^\alpha \mathrm{d}t + \int_0^1 [I_0(t)f(t)]t^\alpha \ln(1/t)\mathrm{d}t$$
$$+ e^{-1}\int_0^\infty [(1+t)^{\alpha-1/2} S(1+t) f(1+t)] e^{-t}\mathrm{d}t$$

example $w(t) = t^\alpha K_0(t)$ on $[0,\infty]$, $\alpha > -1$

$$K_0(t) = \begin{cases} R(t) + I_0(t)\ln(1/t) & \text{if } 0 < t \le 1, \\ t^{-1/2}e^{-t}S(t) & \text{if } 1 \le t < \infty, \end{cases}$$

$$\int_0^\infty f(t)w(t)\mathrm{d}t$$

$$= \int_0^1 [R(t)f(t)]t^\alpha \mathrm{d}t + \int_0^1 [I_0(t)f(t)]t^\alpha \ln(1/t)\mathrm{d}t$$

$$+ e^{-1}\int_0^\infty [(1+t)^{\alpha-1/2}S(1+t)f(1+t)]e^{-t}\mathrm{d}t$$

$$AB = \begin{array}{||c|c||} \hline\hline 0 & 1 \\ 0 & 1 \\ 0 & \infty \\ \hline\hline \end{array}$$

$$w_1(t) = t^\alpha$$

$$w_2(t) = t^\alpha \ln(1/t)$$

$$w_3(t) = e^{-t}$$

Modification Algorithms G., 1982

Modification Algorithms **G., 1982**

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

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

Modification Algorithms G., 1982

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

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

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 G., 1982

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

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

example: Galant, 1971

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

one step of (symmetric) LR algorithm:

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

nonlinear recurrence algorithm

Matlab

`ab=chri1(N,ab0,c)`, `chri2`, . . ., `chri8`

application: induced orthogonal polynomials (G. and Li, 1993)

$$d\lambda_{\text{mod}}(t) = [\pi_m(t; d\lambda)]^2 d\lambda(t)$$

$$[\pi_m(t; d\lambda)]^2 = \prod_{\mu=1}^{m} (t - \tau_\mu)^2$$

m consecutive modifications by quadratic factors

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

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

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

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

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

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

algorithm G. and Zhang, 1995

conditioning Zhang, 1994

Matlab (for $s = 1$)

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

example $\mathrm{d}\lambda_0(t) = \mathrm{d}t, \quad \mathrm{d}\lambda_1(t) = \gamma \mathrm{d}t \quad$ 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)
md=max(nd),   a0=
```
$\alpha_0(d\lambda_0)$

Matlab (cont')

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

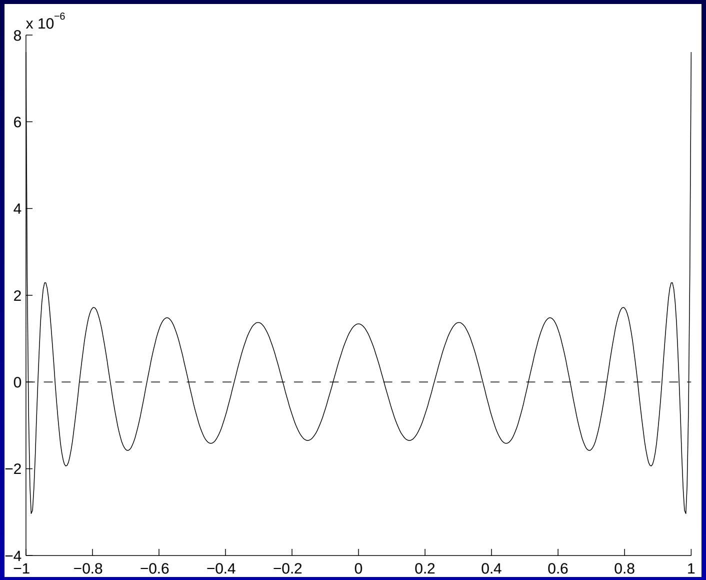
example Althammer's polynomials

Matlab

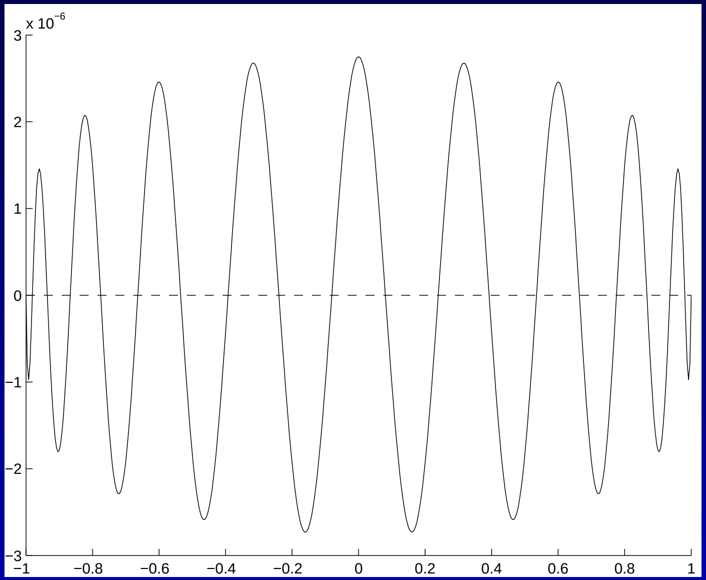
```
s=1; nd=[N N];
a0=0; same=1;
ab=r_jacobi(N);
zw=gauss(N,ab);
xw=[zw(:,1) zw(:,1) ...
     zw(:,2) g*zw(:,2)];
B=stieltjes_sob(N,s,nd,xw,a0,same);
```

Demo #4

Legendre vs Althammer polynomials



Legendre, $n = 20$



Althammer, $n = 20$

Zeros

if

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

then

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

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

Matlab

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