Порождение семейства ортогональных многочленов дискретной переменной для заданного множества узлов

Парфенов Д.В.

Предложен и обоснован вычислительно эффективный метод синтеза семейства дискретных многочленов комплексного аргумента с единичной весовой функцией, удовлетворяющих условию ортогональности на заданном произвольном конечном множестве несовпадающих узлов. Приведён детальный алгоритм расчёта коэффициентов и табуляции значений многочленов на языке GNU Octave/Mathworks Matlab. Дана оценка сложности, показано, что она минимально возможная.

Ключевые слова: цифровая обработка сигналов, устойчивая интерполяция, ортогональные многочлены дискретной комплексной переменной, произвольные узлы, расчёт коэффициентов, оптимальный алгоритм, вычислительная сложность.

В теории и практике современных вычислений заметная роль отводится дискретным ортогональным многочленам. В частности, в задачах устойчивой аппроксимации, построения квадратур, решения дифференциальных уравнений находят применение известные семейства дискретных ортогональных многочленов: Хана, Мейкснера, Кравчука, Шарлье и другие [1]. Многочисленные полезные свойства обеспечивают широкое применение ортогональных многочленов дискретной переменной и в современных алгоритмах цифровой обработки сигналов, например, в цифровой фильтрации, синтезе линейных устройств, при обработке радиосигналов, звука и изображения.

Нередко возникают ситуации, когда конечный набор узлов, на которых следует обеспечить ортогональность, продиктован задачей, нестандартен и фиксирован, а синтезировать соответствующую весовую функцию явным образом нецелесообразно из соображений минимизации вычислительной сложности. В данной работе предлагается простой алгоритм отыскания коэффициентов семейства многочленов степени не вы-

ше D, ортогональных на заданном множестве из $D+1$ различных узлов с единичной весовой функцией, и приводится его вычислительная сложность. Выкладки и программная реализация соответствуют случаю одномерных многочленов комплексной переменной и допускают как упрощение для случая действительной переменной, так и обобщения на многомерный случай и для заданных произвольных весовых функций, выходящие, однако, за рамки данной статьи.

Широко известно (см., например, [1]), что для всех семейств ортогональных многочленов справедлива так называемая трёхчленная рекуррентная формула, связывающая многочлены трёх смежных степеней. В литературе её обычно приводят в виде с тремя коэффициентами, хотя для монарных (т.е. имеющих единичный коэффициент при старшей степени) многочленов существует более простая форма:

$$p_{d+1}(x) = (x - \alpha_d)\, p_d(x) - \beta_d\, p_{d-1}(x). \tag{1}$$

Здесь $p_d(x)$ – ортогональный многочлен степени d переменной x, а α_d и β_d – некоторые комплекснозначные коэффициенты. В дальнейшем переменная x для краткости опущена в обозначениях многочленов. К сожалению, даже в специальной литературе не часто встречаются явные выражения для α_d и β_d, хотя они давно получены (см., например, [2]). Обоснованием приводимого ниже алгоритма служит следующая простая теорема.

Теорема 1. *Для произвольного семейства монарных ортогональных многочленов справедливо соотношение (1), причём:*

$$\alpha_d = \langle p_d, x\, p_d\rangle \,/\, \langle p_d, p_d\rangle\,, \quad \beta_d = \langle p_d, p_d\rangle \,/\, \langle p_{d-1}, p_{d-1}\rangle\,. \tag{2}$$

Доказательство. Введём вспомогательный многочлен $q_d := p_{d+1} - (x - \alpha_d)p_d + \beta_d p_{d-1}$. Действительно, с учётом монарности p_d и p_{d+1}, это многочлен степени d. Вследствие взаимной ортогональности p_{d-1}, p_d и p_{d+1} по условию теоремы, $\langle q_d, p_m\rangle = -\langle p_d, x\, p_m\rangle$, $m = 0, \ldots, d-2$. Но p_d ортогонален всем многочленам степени не выше $d-1$, откуда $\langle q_d, p_m\rangle = 0$. Подстановка (2) в $\langle q_d, p_{d-1}\rangle$ и $\langle q_d, p_d\rangle$ даёт два тождественных нуля. Таким образом, $\langle q_d, p_m\rangle = 0, \quad m = 0, \ldots, d$, но это означает $q_d \equiv 0$ и справедливость (1) при условии (2). □

Теорема 1 лежит в основе организации вычислений предлагаемого алгоритма *одновременного* нахождения коэффициентов и значений ортогональных с единичным весом многочленов на заданном множестве узлов $\{x_m\}_{m=1}^{D+1}$. Первые два монарные многочлена: $p_0 \equiv 1$ и

$p_1 = a_0 + x$, где из требования их взаимной ортогональности находим $a_0 = -(D{+}1)^{-1} \sum_{m=1}^{D+1} x_m$. Остальные многочлены вычисляются последовательно с помощью (2) и (1). Оптимизация учитывает четырёхкратное использование скалярных произведений вида $\langle p_d, p_d \rangle$: при вычислении числителя β_d, знаменателей α_d и β_d (от прошлой итерации) и при нормировке. Нормировка не является обязательной, её введение преследует две цели – уменьшение диапазона значений и повышение точности вычислений с плавающей точкой, что подтверждается экспериментально.

Из соображений компактности и ясности описания алгоритма на Листинге 1 приведён комментированный текст программы на языке систем GNU Octave/Mathworks Matlab[1]. Обозначения в листинге максимально соответствуют использованным в тексте: pdpd := $\langle p_d, p_d \rangle$, pdxpd := $\langle p_d, x\, p_d \rangle$, pd1pd1_inv := $\langle p_{d-1}, p_{d-1} \rangle^{-1}$, pdpd_inv := $\langle p_d, p_d \rangle^{-1}$, alpha_d := α_d, beta_d := β_d. Дополнительно обозначим $\mathbf{x}$ – вектор $D{+}1$ фиксированных различных узлов, на которых следует обеспечить ортогональность. Выходные параметры:

- a – матрица коэффициентов ортонормированных многочленов, где a(i, j) – значение коэффициента при степени $j - 1$ в многочлене степени $i - 1$, $i, j = 1, \ldots, D + 1$,
- s – матрица значений ортонормированных многочленов в узлах $\mathbf{x}$, где s(i, j) – значение многочлена степени $i - 1$ в j-ом узле $i, j = 1, \ldots, D + 1$.

```
a=zeros(D+1,D+1); % треугольная матрица коэффициентов
a(1,1)=1; % единственный ненулевой коэффициент p_0
a0=-sum(x)./(D+1); a(2,1:2)=[a0,1]; % ненулевые коэффициенты p_1
% s - матрица значений многочленов в фиксированных узлах x
s=[ones(1,D+1); a0+x; zeros(D-1,D+1)];

pdpd_inv=1./(D+1);
for d=2:D, % d - степень вычисляемого многочлена
  % находим 2 коэффициента в (1)
  pd1pd1_inv=pdpd_inv; % от предыдущей итерации / входа в цикл
  pdpd=sum(s(d,:).*s(d,:));
  pdpd_inv=1./pdpd;
  pdxpd=sum(s(d,:).*s(d,:).*x);
```

[1]На подобных C++ языках его объём вдвое больше и значительно хуже наглядность.

```
  alpha_d=pdxpd.*pdpd_inv;
  beta_d=pdpd.*pd1pd1_inv;

  % значения вычисляемого многочлена в узлах
  s(d+1,:)=(x-alpha_d).*s(d,:)-beta_d.*s(d-1,:);
  % коэффициенты вычисляемого многочлена
  a(d+1,1)=-alpha_d.*a(d,1)-beta_d.*a(d-1,1); % a_0
  if d>=3, % a_1,...,a_{d-2}
    a(d+1,2:d-1)=
      a(d,1:d-2)-alpha_d.*a(d,2:d-1)-beta_d.*a(d-1,2:d-1);
  end;
  a(d+1,d)=a(d,d-1)-alpha_d.*a(d,d); % a_{d-1}
  a(d+1,d+1)=1.0; % a_d

  % нормировка s(d-1,:) и a(d-1,:)
  % s(d-1,:) и a(d-1,:) не будут использованы для вычисления
  % следующих многочленов; pd1pd1_inv перезапишется в начале
  % следующей итерации
  pd1pd1_inv=sqrt(pd1pd1_inv); % нормирующий множитель
  s(d-1,:)=s(d-1,:).*pd1pd1_inv; % нормируем значения многочлена
  a(d-1,1:d-1)=a(d-1,1:d-1).*pd1pd1_inv; % нормируем коэффициенты
end;
% последний нормированный в цикле многочлен (степени D-2)
% был (D-1)-ым в массивах a и s. Нормируем оставшиеся многочлены
% степени (D-1) и степени D
pdpd_inv=sqrt(pdpd_inv);
s(D,:)=s(D,:).*pdpd_inv; % нормируем значения многочлена
a(D,1:D)=a(D,1:D).*pdpd_inv; % нормируем ненулевые коэффициенты
pdpd_inv=sqrt(1./sum(s(D+1,:).*s(D+1,:)));
s(D+1,:)=s(D+1,:).*pdpd_inv; % нормируем значения многочлена
a(D+1,:)=a(D+1,:).*pdpd_inv; % нормируем коэффициенты
```

Листинг 1.

Фрагмент кода на Листинге 2 иллюстрирует возможный способ разложения вектора $\mathbf{y}$ отсчётов некоторой величины в узлах $\mathbf{x}$ по синтезированным данным способом многочленам.

```
g=zeros(D+1,1); % коэффициенты разложения последовательности
% (y) по ортонормированным многочленам
```

```
for d=1:D+1, g(d)=sum(s(d,:).*y); end;
```

Листинг 2.

Вычисления последних двух строк могут быть выполнены за время не $O(D^2)$, как в приведённом выше иллюстративном примере, а $O(D\ log^2(D))$, если применить методы быстрых полиномиальных преобразований, подобные [3] и [4], но это возможное улучшение касается использования результатов вычислений и не затрагивает существо предлагаемого алгоритма и его временну́ю оценку сложности. Последнюю легко получить из анализа листинга. В главном цикле, выполняющемся $D-1$ раз, на каждой итерации вычисляется $D+1$ значение s многочлена в заданных узлах, таков же порядок сложности нормировки этих отсчётов. Коэффициенты a образуют треугольную матрицу, поэтому объём связанных с ними вычислений уполовинивается: $(D+1)^2/2$. Таким образом, алгоритм имеет квадратическую сложность по D. Очевидно, меньше она быть не может, так как всего вычисляется $(D+1)^2$ значений набора ортогональных многочленов (каждый многочлен в каждом узле). Поскольку в большинстве современных процессоров деление выполняется в несколько раз дольше умножения, число делений минимизировано – всего одно на итерацию.

Список литературы

[1] Никифоров А.Ф., Суслов С.К., Уваров В.Б., Классические ортогональные полиномы дискретной переменной. – М.: Наука, 1985. – 216 с., ил.

[2] Orthogonal Polynomials and Special Functions: Computation and Applications. / Eds. Marcellan F., Van Assche W. – Lecture Notes in Mathematics 1883, Springer, 2006. – 418 p.

[3] Driscoll J., Healy D., Rockmore D., Fast Discrete Polynomial Transforms with Applications to Data Analysis for Distance Transitive Graphs // SIAM J. Comput. Vol. 26, 1996, pp. 1066–1099.

[4] Potts D., Steidl G., Tasche M. - Fast Algorithms for Discrete Polynomial Transforms // Mathematics of Computation, Vol. 67, No 224, Oct. 1998, pp. 1577-1590.

Generation of the ensemble of discrete orthogonal polynomials with a given node set

Parfenov D.V.

A computationally efficient method for generation the ensemble of discrete orthogonal polynomials of complex variable is motivated and proposed. The ensemble keeps orthogonality at a given arbitrary separate node set with unitary weight function. The detailed algorithm implementation for coefficient computation and polynomial sampling is presented in GNU Octave/Mathworks Matlab language. The complexity estimate is the least attainable.

Keywords: digital signal processing, robust interpolation, orthogonal polynomials of discrete complex variable, arbitrary nodes, coefficient computation, optimal algorithm, computational complexity.