УДК 519.688

НОВЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ АППРОКСИМАЦИЙ ПАДЕ И ЕГО РЕАЛИЗАЦИЯ В MATLAB

О.Л. Ибряева

A NEW ALGORITHM FOR CALCULATING PADE APPROXIMANTS AND ITS MATLAB IMPLEMENTATION

O.L. Ibryaeva

В работе предложен алгоритм вычисления аппроксимации Паде, основанный на выборе ее знаменателя с минимальной степенью. Показано, что новый алгоритм не приводит к появлению дуплетов Фруассара, в отличие от имеющихся в Maple и Mathematica процедур вычисления аппроксимаций Паде.

Ключевые слова: аппроксимация Паде, дуплеты Фруассара, метод Паде – Лапласа, плохо обусловленная задача.

A new algorithm for calculating a Pade approximant is proposed. The algorithm is based on the choice of the Pade approximant's denominator of least degree. It is shown that the new algorithm does not lead to the appearance of the Froissart doublets in contrast to available procedures for calculating Pade approximants in Maple and Mathematica.

Keywords: Pade approximant, Froissart doublets, Pade – Laplace method, ill-posed problem.

Введение

Аппроксимации Паде – это локально наилучшие рациональные аппроксимации заданного степенного ряда. Они находят многочисленные применения в различных задачах математической физики, механики и прикладной математики (см., например, [1]). Фактически аппроксимации Паде уже стали неотъемлемой частью научно-технических расчетов, свидетельством чего является создание специальных программ их нахождения в таких системах компьютерной математики как Mathematica и Maple.

Краткий обзор методов вычисления аппроксимаций Паде можно найти в [1], где приведена также программа PADE на Фортране, основанная на решении систем линейных уравнений методом Жордана-Гаусса. При довольно сильных ограничениях для нахождения аппроксимации Паде могут быть использованы и рекуррентные соотношения. В пакете Maple реализованы два алгоритма вычисления аппроксимаций Паде – «быстрый» алгоритм Кабая и Чоя [2] и алгоритм Геддеса [3]. В Matlab, процедура pade находит лишь диагональные аппроксимации Паде функции e^{-st} по имеющимся в данном случае явным формулам.

В настоящей статье предложен новый алгоритм нахождения аппроксимаций Паде, основанный на специальном выборе ее знаменателя и результатах работы [4]. Цель создания такого алгоритма – избавиться от неоднозначности нахождения знаменателя аппроксимации Паде. Разработка алгоритма и его реализация в системе Matlab были мотивированы следующим.

Стандартная программа pade в системе Maple не учитывает многомерности (в общем случае) ядра матрицы, из которого находятся коэффициенты знаменателя аппроксимации

Паде. Это приводит к тому, что числитель и знаменатель аппроксимации Паде, вычисленной с помощью команды pade, могут иметь общие множители. В условиях точных вычислений эти множители сокращаются, и сама аппроксимация Паде находится единственным образом. Однако на практике из-за погрешностей в коэффициентах ряда и ошибок вычислений эти общие множители оказываются лишь примерно равными, что приводит к нежелательному появлению так называемых дуплетов Фруассара.

С дуплетами Фруассара приходится сталкиваться, например, при нахождении параметров A_j, γ_j сигналов вида $f(t) = \sum_{j=1}^{N} A_j e^{\gamma_j t}$ с помощью метода Паде – Лапласа [5]. Этот метод [6] основан на построении преобразования Лапласа для функции $f(t)$, которое в данном случае оказывается рациональной дробью с полюсами γ_j, и его аналитического продолжения с помощью аппроксимации Паде. Для устранения дуплетов Фруассара и определения истинных полюсов γ_j аппроксимации Паде приходится вычислять не только аппроксимацию нужного типа, но и другие «соседние» с ней. Между тем, можно избежать появления дуплетов Фруассара, если сразу находить знаменатель аппроксимации Паде, не содержащий «лишних» нулей (не являющихся полюсами аппроксимируемой рациональной функции). Именно на таком выборе знаменателя и основан предложенный в §3 этой статьи алгоритм вычисления аппроксимации Паде.

Для того, чтобы в дальнейшем иметь возможность анализировать сигналы с помощью метода Паде – Лапласа, этот алгоритм был реализован нами в системе Matlab. Именно Matlab применяется для обработки сигналов особенно эффективно и эффектно, но имеющейся в нем процедуры pade, вычисляющей аппроксимации Паде только в очень частном случае, явно недостаточно для «языка технических вычислений». Описание процедуры, реализующей предложенный алгоритм и примеры ее работы приведены в §4.

1. Основные сведения

Аппроксимации Паде могут быть определены в любой конечной и бесконечно удаленной точке. Дадим определение аппроксимаций Паде в нуле.

Определение 1. *Аппроксимацией Паде типа* (n, m) *для ряда* $c(z)=\sum\limits_{k=0}^{\infty} c_k z^k$ *называется рациональная функция* $\pi_{n,m}(z) = \frac{P_{n,m}(z)}{Q_{n,m}(z)}$ *такая, что многочлены* $P_{n,m}(z)$, $Q_{n,m}(z)$ *удовлетворяют условиям:*

1. $Q_{n,m}(z) \not\equiv 0$, $\deg Q_{n,m}(z) \le m$,
2. $\deg P_{n,m}(z) \le n$,
3. $c(z)Q_{n,m}(z) - P_{n,m}(z) = O(z^{n+m+1})$, $z \to 0$.

Из этого определения следует, что вектор, составленный из коэффициентов знаменателя $Q_{n,m}(z)$, принадлежит ядру теплицевой матрицы

$$T_{n+1} = \begin{pmatrix} c_{n+1} & c_n & \cdots & c_{n-m+1} \\ c_{n+2} & c_{n+1} & \cdots & c_{n-m+2} \\ \vdots & \vdots & & \vdots \\ c_{n+m} & c_{n+m-1} & \cdots & c_n \end{pmatrix}. \tag{1}$$

Ясно, что ядро этой матрицы размером $m \times (m+1)$ всегда ненулевое. С помощью найденного знаменателя $Q_{n,m}(z)$ и условия 3 определения 1 легко находится числитель $P_{n,m}(z)$ аппроксимации. Таким образом, аппроксимация Паде всегда существует. Так как ранг матрицы T_{n+1} может быть меньше m, многочлен $Q_{n,m}(z)$, а значит и $P_{n,m}(z)$, находится, вообще говоря, не единственным образом.

В работе [4] показано, что множество всех знаменателей аппроксимации Паде допускает параметризацию $Q_{n,m}(z) = q(z)Q_1(z)$, где $q(z)$ – произвольный многочлен степени не выше $n - \mu_1$ и μ_1, $Q_1(z)$ – так называемые первый существенный индекс и первый существенный многочлен последовательности $c_{n-m+1}, c_{n-m+2}, \ldots, c_{n+m}$. Таким образом, из всех многочленов $Q_{n,m}(z)$ первый существенный многочлен $Q_1(z)$ имеет минимальную степень. Предложенный в этой статье алгоритм нахождения аппроксимации Паде основан на выборе в качестве знаменателя аппроксимации многочлена $Q_1(z)$.

Множество всех числителей аппроксимации Паде также допускает параметризацию вида $P_{n,m}(z) = q(z)P_1(z)$, где $P_1(z)$ – числитель аппроксимации, соответствующий знаменателю $Q_1(z)$. Корнями многочлена $q(z)$ являются общие ненулевые корни числителя $P_{n,m}(z)$ и знаменателя $Q_{n,m}(z)$ аппроксимации Паде.

На практике, из-за неточных исходных данных и ошибок вычислений, часто оказывается, что «лишнему» корню ν знаменателя (не полюсу аппроксимируемой функции) соответствует лишь примерно равный ему корень $\nu+\varepsilon$ числителя. Такая пара корней носит название дуплетов Фруассара (Froissart doublets). Неизбежное в реальных условиях появление дуплетов Фруассара серьезно затрудняет определение истинных полюсов аппроксимируемой функции.

2. Вычисление аппроксимаций Паде в системах компьютерной математики

В этом параграфе мы рассмотрим, как алгоритм вычисления аппроксимаций Паде реализован в таких популярных математических пакетах, как Maple, Mathematica и Matlab.

В Maple встроенная процедура *pade(f, x = a, [n, m])* позволяет найти аппроксимацию Паде типа (n, m) в точке $x = a$ функции $f(x)$. Если при обращении к процедуре *pade* запросить доступную для пользователя информацию, указав, например, значение параметра *infolevel[pade]:= 1*, то будет получено сообщение об используемом для вычисления аппроксимации алгоритме. В случае, когда коэффициенты ряда Тейлора функции $f(x)$ не содержат параметров и чисел с плавающей точкой, используется «быстрый» алгоритм Кабая и Чоя [2]. В остальных случаях нахождение аппроксимации проводится по алгоритму Геддеса [3], использующему символьные вычисления.

В случае неодномерного ядра матрицы (1) вектор, составленный из коэффициентов знаменателя, может быть найден не надлежащим образом, что может привести к появлению «лишних» нулей знаменателя и числителя. Покажем это на примерах.

Пример 1

Найдем аппроксимацию Паде $\pi_{2,3}(f_1)$ типа $(2, 3)$ функции $f_1(x) = \frac{x+1{,}0001}{(x+1{,}999)(x-2{,}001)}$ в точке $x = 0$. В этом случае ядро матрицы (1) неодномерно, однако, как видно из примера, найденные знаменатель и числитель не содержат «лишних» нулей.

restart;
with(numapprox);
padef1:= pade $\left(\frac{x+1{,}0001}{(x+1{,}999)(x-2{,}001)}, x = 0, [2, 3]\right)$;

$$\frac{-0{,}2500250626-0{,}2500000625x}{1{,}000000000+0{,}0005000002006x-0{,}2500000625x^2}$$

factor(denom(padef1), complex); solve(numer(padef1) = 0);
$0,2500000625(x + 1,999000000)(x - 2,001000000)$
$-1,000100000.$

Пример 2

Найдем аппроксимацию Паде $\pi_{4,5}(f_2)$ типа $(4,5)$ функции $f_2(x) = \frac{(x-3{,}001)(x+1{,}9999)}{(x^2+1)(x+4{,}0001)}$ в точке $x = 0$. В этом случае ядро матрицы (1) неодномерно, и найденные знаменатель и числитель аппроксимации Паде имеют «лишние» нули (они выделены жирным шрифтом).

padef2:= $pade\left(\frac{(x-3{,}001)(x+1{,}9999)}{(x^2+1)(x+4{,}0001)}, x=0, [4,5]\right)$*;*

$$\frac{-1{,}500387465-0{,}3753104091x-0{,}4121913908x^2-0{,}08614086896x^3+0{,}1068576988x^4}{1{,}000000000+0{,}3333333332x+1{,}448275862x^2+0{,}4401910336x^3+0{,}4482758628x^4+0{,}1068577004x^5}$$

factor(denom(padef2), complex); factor(numer(padef2), complex);

$0,1068577004(x+4,000100004)(x+\mathbf{0,09748654036+1,526433054I})$
$(x+\ \mathbf{0,09748654036-1,526433054I})(x+1,000000001I)(x-1,000000001I)$
$0,1068576988(x+1,999900006)(x+\mathbf{0,09748653975+1,526433060I})$
$(x\ +\ \mathbf{0,09748653975-1,526433060I})(x-3,001000018)$.

Отметим, что в обоих приведенных примерах мы находимся в так называемом сингулярном блоке таблицы Паде для аппроксимируемой рациональной функции и, согласно теории аппроксимаций Паде, $\pi_{2,3}(f_1)$, $\pi_{4,5}(f_2)$ должны были оказаться равными рациональным дробям $f_1(x)$ и $f_2(x)$.

Найденная с помощью Maple аппроксимация $\pi_{2,3}(f_1)$ оказалась равной аппроксимируемой функции, и в этом случае не появилось лишних нулей числителя и знаменателя. Однако аппроксимация $\pi_{4,5}(f_2)$ не совпала с $f_2(x)$, у числителя и знаменателя аппроксимации появились близкие нули – дуплеты Фруассара. Дело в том, что из неодномерного ядра матрицы T_{n+1} (1) в первом случае (случайно) был выбран «правильный» знаменатель. Во втором же примере взятый знаменатель не совпал со знаменателем рациональной дроби $f_2(x)$.

Похожая картина наблюдается и при попытке найти $\pi_{2,3}(f_1)$, $\pi_{4,5}(f_2)$ с помощью системы Mathematica. Как показывают приведенные ниже примеры, при этом опять появляются дуплеты Фруассара.

Пример 3

Вновь, как в примере 1, найдем аппроксимацию Паде $\pi_{2,3}(f_1)$ типа $(2,3)$ функции $f_1(x) = \frac{x+1{,}0001}{(x+1{,}999)(x-2{,}001)}$ в точке $x = 0$. Появляющиеся при этом дуплеты Фруассара выделены жирным шрифтом.

In[1]:= PadeApproximant$[\frac{x+1{,}0001}{(x+1{,}999)(x-2{,}001)}, \{x, 0, \{2,3\}\}]$
Out[1]= $\frac{-0{,}250025-0{,}222305x+0{,}0276927x^2}{1{,}000000000000000-0{,}110271x-0{,}250055x^2+0{,}0276927x^3}$
In[2]:= Roots[Denominator[Out[1]] == 0, x]
Out[2]= $x==-1,999||x==2,001||\mathbf{x==9,02765}$
In[3]:= Roots[Numerator[Out[1]] == 0, x]
Out[3]= $x==-1,0001||\mathbf{x==9,02765}$

Пример 4

Как в примере 2, найдем аппроксимацию Паде $\pi_{4,5}(f_2)$ типа $(4,5)$ функции $f_2(x) = \frac{(x-3{,}001)(x+1{,}9999)}{(x^2+1)(x+4{,}0001)}$ в точке $x = 0$.

In[4]:= PadeApproximant$[\frac{(x-3{,}001)(x+1{,}9999)}{((x^2+1)(x+4{,}0001)}, \{x, 0, \{4,5\}\}]$
Out[4]= $\frac{-1{,}50039-25{,}4213x-6{,}51318x^2+3{,}7662x^3+0{,}42731x^4}{1{,}00000000000000+17{,}0263x+6{,}90326x^2+17{,}4536x^3+5{,}90326x^4+0{,}42731x^5}$
In[5]:= Roots[Denominator[Out[4]] == 0, x]
Out[5]:= $\mathbf{x==-9,75487}||x==-4,0001||\mathbf{x==-0,0599743}||x==-1,0I||x==+1,0I$
In[6]:= Roots[Numerator[Out[4]] == 0, x]
Out[6]= $\mathbf{x==-9,75487}||x==-1,9999||\mathbf{x==-0,0599743}||x==3,001$

Итак, числитель и знаменатель найденной по алгоритму, реализованному в Maple и Mathematica аппроксимации Паде, могут оказаться не взаимно простыми. Для устранения приближенных общих множителей можно воспользоваться процедурой EpsilonGCD из пакета SNAP. Однако это может привести к плохо обусловленной задаче и к неоправданной потере точности. Таким образом, нужно реализовать такой алгоритм вычисления аппроксимации Паде, при котором в качестве ее знаменателя выбирается многочлен минимальной степени, не содержащий «лишних» нулей. Как уже было сказано в §1, среди всех знаменателей $Q_{n,m}(z)$ аппроксимации Паде именно первый существенный многочлен $Q_1(z)$ обладает таким свойством.

В следующем параграфе приведен алгоритм вычисления аппроксимации Паде, основанный на выборе в качестве ее знаменателя многочлена $Q_1(z)$. Этот алгоритм был реализован нами в системе компьютерной математики Matlab. Имеющаяся в этой системе процедура pade осуществляет Паде – аппроксимацию передаточной функции звена временного запаздывания, т.е. фактически находит диагональные аппроксимации Паде функции e^{-st} по имеющимся в данном случае явным формулам [7].

3. Алгоритм вычисления аппроксимаций Паде

В этом параграфе мы приводим алгоритм для нахождения аппроксимации Паде типа (n, m) функции $f(x)$ в точке $x = a$:

1. Находим коэффициенты $c_0, \ldots, c_{n+m}$ **разложения в ряд Тейлора** аппроксимируемой функции $f(x) = \sum\limits_{k=0}^{\infty} c_k(x-a)^k$ в точке $x = a$.

2. Составим теплицевы матрицы (здесь $c_k = 0$, если $k < 0$)

$$T_k = \begin{pmatrix} c_k & c_{k-1} & \ldots & c_M \\ c_{k+1} & c_k & \ldots & c_{M+1} \\ \vdots & \vdots & & \vdots \\ c_N & c_{N-1} & \ldots & c_{N+M-k} \end{pmatrix}, \quad M \le k \le N,$$

где $M = n - m + 1$, $N = n + m$.

Ядра этих матриц образуют цепочку вложенных пространств. Первое нетривиальное ядро в этой цепочке обязательно является одномерным, а производящий многочлен его базиса является искомым знаменателем $Q_1(z)$. Тем не менее, мы не ограничимся вычислением ядра для $Q_1(z)$, а найдем ранги всех матриц T_k для того, чтобы иметь возможность сделать проверку правильности результата (см. далее п. 5).

3. Находим ранги r_k **матриц** T_k**,** $M \le k \le N$.

Наиболее надежным способом решения этой некорректной задачи – нахождение ранга матрицы – является метод, основанный на сингулярном разложении. Заметим, что этот метод и реализован в пакете Matlab.

Функция rank(A) в Matlab возвращает ранг матрицы A, который определяется как количество ее сингулярных чисел, превышающих порог tol. Это пороговое значение зависит от размеров матрицы, ее спектральной нормы и относительной точности вычислений eps=2^{-52}. Пользователь также имеет возможность указать свое значение tol при использовании функции r = rank(A, tol) для вычисления ранга матрицы.

Вряд ли можно однозначно решить вопрос о выборе порогового значения (фактически, вопрос о том, какие малые числа уже можно считать нулевыми). Ясно, что это существенно зависит и от прикладной задачи, при решении которой требуется найти аппроксимации Паде, и от точности задания исходных данных. Мы в своей программе оставили значение tol, используемое в Matlab по умолчанию.

4. Находим размерности $d_k = k - M + 1 - r_k,\ M \le k \le N$ **правых ядер матриц** T_k**.** Положим также $d_{M-1} = 0$, $d_{N+1} = N - M + 2$.

5. Составляем разности $\Delta_k = d_k - d_{k-1}$**,** $M \le k \le N+1$ **и находим существенные индексы** μ_1, μ_2.

Для чисел Δ_k, как следует из работы [4], справедливы равенства:

$\Delta_M = \ldots = \Delta_{\mu_1} = 0, \quad \Delta_{\mu_1+1} = \ldots = \Delta_{\mu_2} = 1, \quad \Delta_{\mu_2+1} = \ldots = \Delta_{N+1} = 2.$

Числа μ_1, μ_2, определяемые этими соотношениями, и называются первым и вторым существенным индексами последовательности $c_M, \ldots, c_N$.

Поскольку мы вычисляем ранги всех матриц T_k, то мы можем проверить, что последовательность Δ_k является монотонной. Нарушение монотонности этой последовательности указывает на то, что индексы найдены неверно. Это может быть вызвано, например, плохой обусловленностью матриц семейства T_k, поскольку в этом случае ненулевые сингулярные числа плохо отделяются от нулевых, либо грубой ошибкой в выборе величины порога tol, обсуждаемой в п. 3.

6. Находим вектор $(g_0, g_1, \ldots, g_{\mu_1+1-M})^T$ **из одномерного ядра матрицы** T_{μ_1+1}. Он содержит коэффициенты знаменателя аппроксимации Паде, имеющего минимальную степень (первый существенный многочлен).

7. Находим знаменатель аппроксимации Паде $Q_1(z) = \sum\limits_{k=0}^{\mu_1+1-M} g_k(z-a)^k$.

Заметим, что степень многочлена $Q_1(z)$ может быть формальной. Если $\deg Q_1(z) = s < \mu_1+1-M$, то обычно считают, что $z = \infty$ является корнем $Q_1(z)$ кратности $\kappa = \mu_1+1-M-s$. Из-за приближенных вычислений коэффициенты g_k, $k = s+1, \ldots, \mu_1+1-M$ могут оказаться лишь близкими к нулю. Это приведет к тому, что многочлен $Q_1(z)$ будет иметь κ корней с большой абсолютной величиной.

Пользователь должен иметь априорную информацию о том, могут ли числитель и знаменатель аппроксимации Паде иметь такие большие по модулю корни. Эта априорная информация учитывается с помощью введения дополнительного параметра $tolerance$, задаваемого пользователем при вызове процедуры для нахождения аппроксимации Паде. Этот параметр используется программой для того, чтобы определить какие из коэффициентов g_k следует считать равными нулю.

Заметим, что приравнивание к нулю коэффициентов g_k, $k = 0, \ldots, s$, меньших чем $tolerance$, не оказывает существенного влияния на корни многочлена в силу их непрерывной зависимости от коэффициентов многочлена.

8. Составляем матрицу $M = \|c_{i-j}\|_{\substack{i=1,\ldots,n+1,\\ j=1,\ldots,s+1,}}$ **(**$c_k = 0$**, если** $k < 0$**), необходимую для нахождения числителя аппроксимации Паде.**

Здесь s – степень многочлена $Q_1(z) = \sum\limits_{k=0}^{s} q_k(z-a)^k$. После исключения почти нулевых коэффициентов в многочлене $Q_1(z)$ его степень s может оказаться меньше формальной степени $\mu_1 + 1 - M$.

9. Находим вектор $(p_0, \ldots, p_n)^T = M \cdot (q_0, \ldots, q_s)^T$ **и числитель аппроксимации Паде** $P_1(z) = \sum\limits_{k=0}^{n} p_k(z-a)^k$. На этом этапе также целесообразно убрать нулевые (меньшие, чем $tolerance$) элементы p_k перед образованием числителя.

4. Описание процедуры

В этом параграфе мы приведем примеры обращения к написанной нами в Matlab процедуре `mypade(f,n,m,a,tolerance)`, которая позволяет по указанному выше алгоритму найти аппроксимацию Паде типа (n, m) функции $f(x)$ в точке $x = a$. Значение параметра $tolerance$

(см. п. 7 алгоритма) указывает, какие малые коэффициенты пользователь считает равными нулю.

Процедура возвращает числитель P и знаменатель Q аппроксимации Паде. Напомним, что многочлен в Matlab представляется в виде вектора, элементами которого являются коэффициенты многочлена, расположенные в порядке убывания степеней.

Пример 5

Приведем пример нахождения с помощью `mypade` аппроксимации Паде $\pi_{2,3}(f_1)$ типа $(2,3)$ в точке $x=0$ функции $f_1(x)=\frac{x+1{,}0001}{(x+1{,}999)(x-2{,}001)}$. Напомним, что в примерах 1 и 3 эта аппроксимация была найдена с помощью пакетов Maple и Mathematica.

Команда `roots` используется для нахождения корней многочленов.

```
>> syms x
>> f=(x+1,0001)/((x+1,999)*(x-2,001));
>> [P,Q]=mypade(f,2,3,0,10^(-10))
P =
 0,24253565356993 -0,24255990713529
Q =
  -0,24253565356993 0,00048507130714 0,97014237174408
>> roots(P)
ans =
  -1,00010000000000
>> roots(Q)
ans =
  2,00100000000000
  -1,99900000000000
```

Итак, корни знаменателя и числителя найденной с помощью нового алгоритма аппроксимации Паде совпали с корнями знаменателя и числителя аппроксимируемой функции. Дуплеты Фруассара не появились.

В приведенном примере значение параметра *tolerance* равно 10^{-10}. Заметим, что тот же результат был получен нами при значениях $tolerance = 10^{-n}$, $n = 4,\dots,16$. Однако при вызове процедуры `mypade` со значением $tolerance = 10^{-17}$ была получена аппроксимация $\frac{-2{,}374402757743255\cdot 10^{-17}z^2-0{,}24253565356993z-0{,}24255990713529}{-0{,}24253565356993z^2+0{,}00048507130714z+0{,}97014237174408}$ с лишним корнем числителя $-1,021459\cdot 10^{16}$. Появление такого большого по модулю корня в рассматриваемой задаче не предполагается и это указывает на то, что значение параметра *tolerance* должно быть увеличено.

Пример 6

Найдем с помощью `mypade` аппроксимацию $\pi_{4,5}(f_2)$ типа $(4,5)$ функции $f_2(x) = \frac{(x-3{,}001)(x+1{,}9999)}{(x^2+1)(x+4{,}0001)}$ в точке $x=0$. Напомним, что при нахождении $\pi_{4,5}(f_2)$ с помощью пакетов Maple и Mathematica в примерах 2 и 4 появлялись дуплеты Фруассара. Однако сейчас, как видно из этого примера, они отсутствуют.

```
>> syms x
>> f=((x-3,001)*(x+1,9999))/((x^2+1)*(x+4,0001));
>> [P,Q]=mypade(f,4,5,0,10^(-10))
P =
  -0,17149454997366 0,17168319397863 1,02925882342746
Q =
  -0,17149454997366 -0,68599534934964 -0,17149454997366 -0,68599534934964
>> roots(P)
```

```
ans =
  3,00100000000000
  -1,99990000000000
>> roots(Q)
ans =
  -4,00010000000000
  0,00000000000000 + 1,00000000000000i
  0,00000000000000 - 1,00000000000000i
```

Итак, при нахождении аппроксимаций $\pi_{2,3}(f_1)$, $\pi_{4,5}(f_2)$ с помощью описанного выше алгоритма не наблюдается появления дуплетов Фруассара.

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации.

Литература

1. Baker, G.A. Pade Approximants / G.A. Baker, P. Graves-Morris. – University Press. – Cambridge, 1996.
2. Cabay, S. Algebraic computations of scaled Pade fractions / S. Cabay, D.K. Choi // SIAM J. on Computing. – 1986. – V. 15, №1. – P. 243 – 270.
3. Geddes, K.O. Symbolic computation of Pade approximants / K.O. Geddes // ACM Transactions on Mathematical Software. – 1979. – V. 5, №2. – P. 218 – 233.
4. Адуков, В.М. Задача аппроксимации Паде как краевая задача Римана / В.М. Адуков // Весцi НАН Беларусi. Серия Фiзiка-матэм. навук. – 2004. – №4. – С. 55 – 61.
5. О дуплетах Фруассара в методе Паде – Лапласа / А.Л. Шестаков, В.М. Адуков, О.Л. Ибряева, А.С. Семенов // Тезисы докладов международной конференции «Системы компьютерной математики и их приложения». – Смоленск, 2011. – С. 257 – 259.
6. Claverie, P. The representation of functions through the combined use of integral transforms and Pade approximants: Pade – Laplace analysis of functions as sums of exponentials / P. Claverie, A. Denis, E. Yeramian // Computer Physics Reports. – 1989. – №9. – P. 247 – 299.
7. Golub, G.H. Matrix Computations / G.H. Golub, C.F. Van Loan. – University Press. – Baltimore, 1989. – P. 557 – 558.

References

1. Baker G.A. *Pade Approximants*. Cambridge, University Press, 1996.
2. Cabay S., Choi D.K. Algebraic computations of scaled Pade fractions. *SIAM J. on Computing*, 1986, v. 15, no. 1, pp. 243 – 270.
3. Geddes K.O. Symbolic computation of Pade approximants *ACM Transactions on Mathematical Software*, 1979, v. 5, no. 2, pp. 218 – 233.
4. Adukov V.M. The problem of Pade approximation as the Riemann boundary problem [Zadacha approksimatsii Pade kak kraevaya zadacha Rimana]. *Vestsi NAN Belarusi. Seriya Fiziko-matem. nauk* [Proceedings of NAS of Belarus. Series of physical and mathematical sciences], 2004, no. 4, pp. 55 – 61.

5. Schestakov A.L., Adukov V.M., Ibryaeva O.L., Semenov A.S. On Froissart doublets in Pade – Laplace method [O dupletakh Fruassara v metode Pade - Laplasa]. *Tezisy dokladov mezhdunarodnoi conferensii «Sistemy compyuternoi matematiki i ikh prilozheniya» [Theses of reports of the International Conference* « Systems of computer mathematics and their applications»], Smolensk, 2011, pp. 257 – 259.
6. Claverie P., Denis A., Yeramian E. The representation of functions through the combined use of integral transforms and Pade approximants: Pade – Laplace analysis of functions as sums of exponentials *Computer Physics Reports*, 1989, no. 9, pp. 247 – 299.
7. Golub G.H., Van Loan C.F. *Matrix Computations.* Baltimore, University Press, 1989, pp. 557 – 558.

Ольга Леонидовна Ибряева, кандидат физико-математических наук, кафедра «Дифференциальные уравнения и динамические системы», Южно-Уральский государственный университет (Россия, г. Челябинск), oli@6v6power.ru

Olga Leonidovna Ibryaeva, Candidate of Physico-mathematical Sciences, Department of Differential Equations and Dynamical Systems, South Ural State University (Russia, Chelyabinsk), oli@6v6power.ru.

Поступила в редакцию 14 июня 2011 г.