УДК 517.9

НЕЛОКАЛЬНЫЙ БИФУРКАЦИОННЫЙ АНАЛИЗ ДВОЙНОГО СГ-УРАВНЕНИЯ МОДИФИЦИРОВАННЫМ МЕТОДОМ ЛЯПУНОВА—ШМИДТА

А. А. Долженков

Воронежский государственный университет

В данной работе рассмотрен подход к нелокальному изучению класса бифуркационных задач вариационного исчисления, включающего в себя большинство эталонных задач солитонной математики и их естественных обобщений, выходящих за рамки этой науки. Основной пример — двойное СГ-уравнение. В основе предложенного подхода — модифицированный метод Ляпунова—Шмидта и методы символьных и численных вычислений, реализованных в системе компьютерной математики Maple.

КЛЮЧЕВЫЕ СЛОВА: бифуркация, метод Ляпунова—Шмидта, двойное СГ-уравнение.

В 70—80-х годах прошлого столетия наблюдалось бурное развитие исследований зарождения волновых процессов в многочисленных теоретических и экспериментальных работах по нелинейной динамике волн [1]. Наибольший прогресс был достигнут в рамках так называемой солитонной математики. В настоящее время, в связи с постоянным увеличением производительности компьютерной техники и совершенствованием программного обеспечения, появились новые возможности в математическом анализе зарождения и развития нелинейных волновых процессов. В данной работе рассмотрен подход к нелокальному изучению класса бифуркационных задач вариационного исчисления, включающего в себя большинство эталонных задач солитонной математики и их естественных обобщений, выходящих за рамки этой науки. Основной пример в работе — двойное СГ-уравнение, но, разумеется, предложенная численная процедура «шире» этого уравнения. В основе предложенного подхода — модифицированный метод Ляпунова—Шмидта [2—6] и методы символьно-численных вычислений, реализованных в системе компьютерной математики Maple. Предложенный подход можно применять с минимальными затратами машинного времени. Так, например, используя двухядерные процессоры и новые версии Maple, можно разделить на разные ядра вычислительные процедуры внутри алгоритма (на базе рассмотренного подхода), что приводит к значительной экономии вычислительных затрат.

1. КРАЕВАЯ ЗАДАЧА ДЛЯ ДВОЙНОГО СГ-УРАВНЕНИЯ

В настоящей статье представлен материал, полученный применением алгоритма (описание программной части дано в среде Maple), с помощью которого становится возможным проведение нелокального анализа существования и свойств решений двойного СГ-уравнения (уравнения sine-Гордон) [1]

$$u_{tt} - u_{xx} = \sin(u) + \frac{1}{2}\sin(2u), \qquad (1)$$

не входящего в совокупность эталонных уравнений солитонной математики, так как оно не поддается исследованию методами этого направления математики. Это уравнение можно весьма успешно исследовать методом Ляпунова—Шмидта и другими методами нелинейного функционального анализа (разумеется, развитый подход применим и в случае обычного СГ-уравнения).

Заметим, что двойные, тройные и n-кратные СГ-уравнения появляются при моделировании волновых явлений в нелинейной оптике (резонансное распространение оптических импульсов), в биологии (одномерные цепочки органических полимеров), при изучении жидкого гелия и т. д. [2]. Первое численное изучение кратного СГ-уравнения было проведено, повидимому, в работах J. C. Eilbeck (1981 г., см. ссылки в [2]).

Если искать решение уравнения (1) в стандартном виде типа «бегущей волны» $u(x,t) = p(kx + \omega t)$, то получим ОДУ второго порядка

$$p'' + \lambda\left(\sin(p) + \frac{1}{2}\sin(2p)\right) = 0, \qquad (2)$$

которое нужно решать при тех же условиях, что и в случае простого СГ-уравнения. Ограничимся, простоты ради, рассмотрением лишь периодических решений, получаемых при краевых условиях первого рода

$$p(0) = p(1) = 0. \qquad (3)$$

Немного обобщив, расширим уравнение (2) до неоднородного уравнения

$$p'' + \frac{\lambda}{2}\left(\sin(p) + \frac{1}{2}\sin(2p)\right) = \sum_{j=1}^{n} \xi_j e_j, \qquad (4)$$

$$e_j = \sqrt{2}\sin(j\pi x).$$

Аналогичное рассуждение можно провести и для исходного уравнения (1).

От полученной краевой задачи можно перейти к «эвивалентной» экстремальной задаче

$$V(p, \lambda, q) \to extr,\ q = (q_1, \ldots, q_n), \qquad (5)$$

где

$$V(p, \lambda, q) = \int_0^1 \left(\frac{1}{2}\left(\frac{dp}{dx}\right)^2 + \right.$$

$$\left. + \lambda\left(\cos(p) + \frac{1}{4}\cos(2p) - \frac{5}{4}\right) + (\sum_{j=1}^{n} q_j e_j)p\right)dx. \qquad (6)$$

2. НЕЛОКАЛЬНАЯ ВАРИАЦИОННАЯ ВЕРСИЯ МЕТОДА ЛЯПУНОВА—ШМИДТА

Гладкое отображение $f : E \to F$ называется фредгольмовым, если его производная Фреше $\frac{\partial f}{\partial x}(x)$ — фредгольмов оператор в каждой точке $x \in E$. Индексом фредгольмова отображения f называется индекс его производной $\frac{\partial f}{\partial x}(x)$:

$$indf := ind\frac{\partial f}{\partial x}(x) := \dim Ker\frac{\partial f}{\partial x}(x) -$$

$$- \dim Coker\frac{\partial f}{\partial x}(x)$$

(индекс $\frac{\partial f}{\partial x}(x)$ не зависит от x). Далее будем считать, что f является фредгольмовым отображением нулевого индекса, и наряду с этим выполнены следующие условия:

а) $E \subset F \subset H$ — тройка непрерывно вложенных банаховых пространств (H — гильбертово пространство);

б) E плотно в H (это означает, что любой элемент из H может быть представлен как предел последовательности из E).

Из плотности E в H следует, что F плотно в H. Уравнение

$$f(x) = 0,\ x \in \mathcal{U} \qquad (7)$$

называется фредгольмовым.

Если для f существует такой гладкий функционал V на E, что $f = grad_H V$ или, что эквивалентно,

$$\frac{\partial V}{\partial x}(x)h = \langle f(x), h\rangle_H,\ \forall x, h \in E$$

($\langle\cdot,\cdot\rangle$ — скалярное произведение в гильбертовом пространстве H), то отображение f называется потенциальным, а функционал V называется потенциалом отображения f. Если V является потенциалом f, то уравнение (7) можно переписать в виде

$$grad_H V(x) = 0,\ x \in \mathcal{U}. \qquad (8)$$

Оно называется уравнением Эйлера—Лагранжа экстремалей (критических точек) функционала V. Точка 0 называется критической для функционала V, если

$$\frac{\partial V}{\partial x}(a)h = \langle f(a), h\rangle_H = 0,\ \ \forall h \in E \setminus \{0\}.$$

Плотность E в H обеспечивает равносильность последнего равенства уравнению (8). То есть построение решений уравнения (7) можно заменить построением критических точек функционала V (вариационный метод). Функционал V называется фредгольмовым, если его градиент — фредгольмово отображение. Критическая точка a функционала V называется невырожденной (морсовской), если

$$\frac{\partial f}{\partial x}(a)h \neq 0,\ \ \forall h \in E \setminus \{0\}.$$

Индексом Морса невырожденной критической точки 0 функционала V называется максимальная размерность подпространства, на котором отрицательно определен второй дифференциал $\frac{\partial^2 V}{\partial x^2}(a)(h, h)$.

Важным типом уравнений, часто встречающимся в задачах математической физики, является фредгольмово уравнение с параметром:

$$f(x, \delta) = b,\ x \in X,\ b \in Y,\ \delta \in \mathbb{R}^k.$$

С изменением параметра (при некоторых критических значениях δ) совокупность решений может количественно изменяться. Если отображение $f(\cdot, \delta)$ потенциально с потенциалом $V(\cdot, \delta)$, то потенциал также гладко зависит от данного параметра. В такой ситуации естественным образом возникает понятие бифуркации экстремалей и, соответственно, бифуркационного значения параметра.

В процессе перехода δ через критическое значение δ_0 говорят о рождении критических

точек из точки a. Очень часто полагают $a = 0$ и рассматривают рождение критических точек из нуля.

Пусть задана гладкая фредгольмова развертка

$$f(\cdot,\delta) : E \to F,\ \delta \in \Delta^m \subset \mathbb{R}^m.$$

Пусть Ω — открытое подмножество в E. Дискриминантным множеством $\Sigma(\Omega)$ уравнения

$$f(x,\delta) = b, x \in \Omega \qquad (9)$$

называется совокупность тех значений $\delta = \overline{\delta}$, для которых данное уравнение имеет в Ω вырожденное решение $\overline{x}$.

Если для потенциального отображения $f : E \to F$ выполнено условие положительности (монотонности)

$$\left\langle \frac{\partial f}{\partial x}(x)h, h \right\rangle > 0 \quad \forall (x,h) \in E \times (E \setminus 0), \quad (10)$$

то, как легко проверить, уравнение $f(x) = 0$ имеет не более одного решения. Это решение является точкой минимума V на E.

В случае выполнения условия собственности f (компактность прообраза произвольного компакта) уравнение $f(x) = 0$ однозначно разрешимо (следствие теоремы Банаха—Мазура—Каччиополи [1]). Его решение является точкой глобального минимума V.

Соотношение (10) можно заменить более слабым условием

$$\left\langle \frac{\partial f}{\partial x}(x)h, h \right\rangle > 0 \quad \forall (x,h) \in E \times (\tilde{E} \setminus 0), \quad (11)$$

где $\tilde{E} = E \cap N^{\perp}, N = Lin(e_1,\ldots,e_n), N^{\perp}$ — ортогональное дополнение к N в H, $e_1,\ldots,e_n$ — некоторая ортонормированная в H система векторов в E. При этом условии можно определить ключевую функцию Ляпунова – Шмидта

$$W(\xi) := \inf_{x:\langle x,e_j\rangle=\xi_j \forall j} V(x),\ \xi = (\xi_1,\ldots,\xi_n)^{\top}, \quad (12)$$

«отвечающую» за поведение функционала V. Условие собственности f можно ослабить, заменив его условием собственности при каждом ξ «послойного» отображения

$$\tilde{f}_{\xi} : \tilde{E} \to \tilde{F}, \qquad (13)$$

где

$$\tilde{F} = F \cap N^{\perp},$$

$$\tilde{f}_{\xi}(v) := P_{\tilde{F}}\left(f(l(\xi)+v)\right) =$$

$$= f(l(\xi)+v) - \sum_{j=1}^{n} \left\langle e_j, f(l(\xi)+v) \right\rangle e_j, \qquad (14)$$

$$l(\xi) = \sum_{j=1}^{n} \xi_j e_j.$$

При выполнении условий (11), (13) уравнение

$$\tilde{f}_{\xi}(v) = q \qquad (15)$$

однозначно разрешимо при всех ξ, q, и его решение $v = \Phi(\xi)$ гладко зависит от ξ — по теореме о неявной функции. Левую часть (12) можно представить в виде

$$W(\xi) \equiv V(l(\xi) + \Phi(\xi)). \qquad (16)$$

Для ключевого уравнения

$$\theta(\xi) = 0,\ \xi \in \mathbb{R}^n, \qquad (17)$$

в котором

$$\theta(\xi) = (\theta_1(\xi),\ldots,\theta_n(\xi))^{\top},$$

$$\theta_j(\xi) = \left\langle f(l(\xi)+\Phi(\xi)), e_j \right\rangle,$$

имеем

$$\theta(\xi) = \operatorname{grad} W(\xi).$$

Впервые условие собственности в нелокальной редуцирующей схеме было использовано Ю. И. Сапроновым. Приведем одно из сформулированных им утверждений (см. [5, 6]): *Пусть отображение (13) является собственным и пусть при этом выполняется условие положительности (11). Тогда маргинальное отображение* $\varphi : \xi \mapsto l(\xi) + \Phi(\xi)$, *где* $\Phi(\xi)$ *определено уравнением (15), устанавливает взаимно однозначное соответствие между критическими точками ключевой функции (12) и заданного функционала* V. *При этом локальные кольца особенностей*[1] *соответствующих функций в точках* ξ *и* $\varphi(\xi)$ *изоморфны, а в соответствующих друг другу однократных критических точках имеет место совпадение индексов Морса.*

Условие собственности отображений (13) можно заменить на любое другое, гарантирующее существование условных экстремалей в слоях $p^{-1}(\xi)$:

$$p(x) = (p_1(x),\ldots,p_n(x))^{T}.$$

Например, если пространство E рефлексивно, то достаточно потребовать коэрцитивность V (наряду с выпуклостью) вдоль каждого слоя. В этих же целях можно применять и известное условие (C) Пале—Смейла.

[1] Локальное кольцо особенности гладкого функционала V в критической точке a определяется как фактор кольца ростков гладких функционалов в точке a по идеалу, порожденному функционалами вида $\alpha(f(x))$, где α — произвольный гладкий функционал, заданный на произвольной окрестности нуля в пространстве F ($f = \operatorname{grad}_H V$).

При $\lambda \le (n+1)^2\pi^2$ для потенциала рассматриваемого уравнения примером глобально редуцирующей системы функционалов является система $p_j(x) = \langle e_j, x\rangle, j = 1,\dots,n$ (схема Ляпунова—Шмидта).

Возможность ее применения обосновывается через рассмотрение более широкого класса задач, описывающих поведение целого ряда потенциальных физических систем:

$$V_{(\lambda,q)}(x) = \int_0^1 \left(\frac{|\dot{x}|^2}{2} - \lambda \frac{|x|^2}{2} + \omega(x,\lambda) - qx \right) dt,$$

$$x : [0,1] \to \mathbb{R}^n,\ \omega''(x,\lambda)(h,h) \ge 0 \quad \forall x, \lambda. \quad (18)$$

Уравнение Эйлера—Лагранжа этого функционала имеет вид

$$\ddot{x} + \lambda x - \frac{\partial \omega}{\partial x}(x,\lambda) = q,$$

где левая часть будет фредгольмовым отображением индекса 0 из

$$E = \left\{ x \in C^2_{[0,1]} : x(0) = x(1) = 0 \right\}$$

в $F = C_{[0,1]}$ и градиентом функционала относительно $H = L_2([0,1])$.

Из неравенства Пуанкаре—Стеклова—Виртингера

$$\int_0^1 |\dot{u}|^2\, dt \ge \pi^2 (k+1)^2 \int_0^1 |u|^2\, dt,$$

верного на $C^1_{[0,1]}$ при $\int_0^1 \sin \pi j t u(t) dt = 0 (j = 1,\dots,n)$, следует выпуклость функционала в слоях $p^{-1}(\xi)$, а выпуклость и коэрцитивность гарантируют существование невырожденной точки абсолютного минимума $\varphi(\xi)$ для $V_\delta(\cdot)|_{p^{-1}(\xi)}$ (однозначность разрешимости $f(x) = 0$ при фиксировании ключевых параметров).

Если представить левую часть уравнения как

$$f(x) = f_1(x) + f_2(x) = \left[\frac{d^2x}{dt^2} + \lambda x \right] + \left[-\frac{\partial \omega}{\partial x}(x,\lambda) \right],$$

то схема Ляпунова—Шмидта сводится в операторном виде к системе:

$$\begin{cases} f_1(u) - P_{E_n}\left(\dfrac{\partial \omega}{\partial x}(u+v,\lambda) \right) = q_1 e_1 + \dots + q_n e_n, \\ f_1(v) - P_{E_{\infty-n}}\left(\dfrac{\partial \omega}{\partial x}(u+v,\lambda) \right) = 0. \end{cases} \quad (19)$$

Для второго уравнения системы имеет место однозначная разрешимость по v при всяком u ($u + v \in E_n \times E_{\infty-n}$). Обозначив решение $v = \Phi(u)$, получим ключевую функцию в виде $W(u) = V(u + \Phi(u))$.

3. ОПИСАНИЕ АЛГОРИТМА (ТЕОРЕТИЧЕСКАЯ ЧАСТЬ)

Для вариационной задачи (5) простейшим приближением служит классическая (линейная) ритцевская аппроксимация

$$\widetilde{V}(\xi,\lambda,q) \to extr,\ \xi \in \mathbb{R}^n,$$

$$\widetilde{V}(\xi,\lambda,q) := V\Big(\sum_{j=1}^n \xi_j e_j, \lambda, q\Big), \quad (20)$$

дающая достаточно хорошую информационную точность в локальных вопросах (типа описания процесса зарождения волн) [5, 6]. В нелокальных же вопросах требуется более точное приближение, основанное на нелинейных ритцевских аппроксимациях [5, 6]

$$W(\xi,\lambda,q) := V\Big(\sum_{j=1}^n \xi_j e_j + \Phi(\xi), \lambda, q\Big),$$

$$\xi \in \mathbb{R}^n,\ \lambda < (n\pi)^2.$$

Нелинейная добавка $\Phi(\xi)$ строится на основе вспомогательной экстремальной задачи

$$V\Big(\sum_{j=1}^n \xi_j e_j + v\Big) \to \inf_v,\ v \perp e_j \quad (21)$$

то есть

$$\Phi(\xi) := arg_v V\Big(\sum_{j=1}^n \xi_j e_j + v\Big).$$

Следствием этих заключений является следующее утверждение: *при* $\lambda < \frac{(n+1)^2\pi^2}{2}$ *существует взаимно однозначное соответствие между критическими точками функционала (6) и функции (21), сохраняющее их индексы Морса и локальные кольца особенностей*

Математическое обоснование и соответствующие примеры имеются в [6].

Ниже рассмотрен лишь случай $n = 2$.

Алгоритм состоит из следующих частей:

1) построение нелинейной добавки $\Phi(\xi)$ (на основе вспомогательной экстремальной задачи (21));

2) построение ключевой функции $W(\xi,\lambda,q)$;

3) построение графических изображений каустики (т. е. множества Σ, состоящего из тех значений q, при которых существуют вырожденные экстремали);

4) построение графических изображений ключевой функции;

5) построение графических изображений отдельных экстремалей (решений исходной краевой задачи и исходного уравнения).

4. ПРОГРАММНАЯ ЧАСТЬ АЛГОРИТМА

В данной части будут опущены части программы, не носящие информативный характер, например такие, как ввод, чтение файлов и др.

```
> restart:with(Student[Calculus1]):with(plots):
```

Шаг для численного дифференцирования и число отрезков разбиения для численного интегрирования

```
> h:=0.001:int_part_count:=60:
```

Параметр численного решения дифференциальных уравнений выявлен опытно, и необходимо иногда увеличивать его от значения по умолчанию.

```
> AErr:=Float(1,-7):
```

Базис пространства

```
> e:=evalf([seq(sqrt(2)*sin(Pi*k*t),k = 1 .. 10)]);
> lambda:=evalf(4*Pi2/2+3);
```

Задание оператора уравнения. Здесь и далее процедурная обертка (оформление через proc) используется потому, что результат численного решения дифференциальных уравнений имеет тип proc, и для того, чтобы его можно было складывать с другими функциями или осуществлять композицию, надо, чтобы все операнды имели такой же тип.

Нелинейная часть оператора уравнения

```
> fn:=x->lambda*(sin(x)+1/2*sin(2*x)):
```

Процедурная обертка нелинейной части

```
> fnp:=proc(x)proc(s)eval(fn(x),t=s);end;end;
```

Оператор уравнения и интегранд его потенциала

```
> f:=x->diff(x,t$2)+fn(x):
> iV:=(x,q)->diff(x,t)2/2+lambda*(cos(x)-1)+
+lambda*1/4*(cos(2*x)-1)+ q)*x:
```

Далее следующим образом осуществляется переход к операторному уравнению с сжимающим оператором.

```
> Diff(x,t$2)+'f'(x)=0,G=(Diff(x,t$2))('-1');
Diff(x,t$2)=y,x=G(y);y+'f'(G(y))=0;
```

Поскольку решение уравнения ищется не на всем пространстве, то реально мы имеем дело со следующими представлениями и решением следующего уравнения:

```
>x=u+v,u=xi[1]*'e'[1]+xi[2]*'e'[2];y=y(u)+y(v),
y=-Pi2*xi[1]*'e'[1]-4*Pi2*xi[2]*'e'[2]+y(v);
> y(v)+P[E[infinity-'2']]*'f'(u+G(y(v)))=0;
```

Численная реализация (как решение дифференциального уравнения) оператора обращения двойного дифференцирования с краевыми условиями

```
> G:=y->rhs(dsolve(diff(x(t),t$2)=y,x(0)=0,x(1
)=0,x(t),numeric,method=bvp,abserr=AErr,
maxmesh=256, output=listprocedure)[2]):
```

Процедурная реализация y(u)

```
> u:=proc(x,y)
> proc(s)eval(x*e[1]+y*e[2],t=s);end;
> end;
```

Процедурная обертка проекторов на

```
> e[1]; и
> e[2]; для процедурных функций (интегралы
```

берутся по отрезку [0,1-h], поскольку вторая производная будет считаться через правую разность).

```
> projector1:=proc(x)
> evalf(ApproximateInt(x(t)*e[1],t=0..1-h,
method = trapezoid, partition=int_part_count));
> end:
> projector2:=proc(x)
> evalf(ApproximateInt(x(t)*e[2],t=0..1-h,
method = trapezoid, partition=int_part_count));
> end:
```

Процедурная обертка проектора на

```
> E[infinity-'2']; для процедурных функций
> proj:=proc(x)
> local p1,p2:
> p1:=projector1(x):
> p2:=projector2(x):
> x-proc(s)eval(p1*e[1]+p2*e[2],t=s);end;
> end:
```

Начальный шаг для всех итерационных процессов в дальнейшем и тест правильности работы последнего проектора

```
> start:=proc(s)eval(e[3],t=s);end;
> proj(start)(1/2);
```

Процедурная реализация итерационного процесса для уравнения

```
> y(v)+P[E[infinity-'2']]*f(G(y)) = 0;.
> Iter:=proc(x,y,st)
> local it:
> it:=u(x,y)+G(st(t)):
> proj(-fnp(it(t)));
> end;
```

Реализация нескольких итераций и получение v (x=u+v).

```
> Reduction:=proc(x,y,st,itcount)
> local z,i:
> z:=st:
> for i from 1 to itcount do
> z:=Iter(x,y,z):
> od:
```

```
> G(z(t));
> end;
```

Слагаемое u представления x=u+v

```
> _u:=proc(x,y)
> proc(s)eval(evalf(x)*e[1]+evalf(y)*e[2],t=s);
end;
> end;
```

Вычисление гессиана ключевой функции в точке

```
> Hessian:=proc(x,y)
> local Red,dx,dy,f,diffp:
```

Вычисляем производные ключевой функции как проекции градиента:

1. Вычисление v

```
> Red:=Reduction(x,y,start,2):
```

2. Вычисление второй производной от v

```
> diffp:=proc(s)
> (Red(s+h)-2*Red(s+h/2)+Red(s))/(h/2)2;
> end:
```

3. Вычисление градиента без учета $\frac{\partial^2}{\partial(t)^2}u$

```
> Diff(u,'$'(t,2));.
> f:=fnp(_u(x,y)(t)+Red(t))+diffp:
```

4. Вычислений проекций и учет $\frac{\partial^2}{\partial(t)^2}u$

```
> Diff(u,'$'(t,2));.
> dx[1]:=projector1(f)-evalf(Pi2)*x:
> dy[1]:=projector2(f)-evalf(4*Pi2)*y:
```

Повторение вычислений с приращением по первой переменной

```
> Red:=Reduction(x+h,y,start,2):
> diffp:=proc(s)
> (Red(s+h)-2*Red(s+h/2)+Red(s))/(h/2)2;
> end:
> f:=fnp(_u(x+h,y)(t)+Red(t))+diffp:
> dx[2]:=projector1(f)-evalf(Pi2)*(x+h):
> dy[2]:=projector2(f)-evalf(4*Pi2)*y:
```

Повторение вычислений с приращением по первой переменной

```
> Red:=Reduction(x,y+h,start,2):
> diffp:=proc(s)
> (Red(s+h)-2*Red(s+h/2)+Red(s))/(h/2)2;
> end:
> f:=fnp(_u(x,y+h)(t)+Red(t))+diffp:
> dx[3]:=projector1(f)-evalf(Pi2)*x:
> dy[3]:=projector2(f)-evalf(4*Pi2)*(y+h):
```

Вычисление гессиана, построение параболического множества

```
> (dx[2]-dx[1])/h*(dy[3]-dy[1])/h-(dy[2]-
-dy[1])/h*(dx[3]-dx[1])/h;
> end;
> pl:=implicitplot(Hessian,0..5,0..5,grid=
=[70,70]):pl;
> file_:=fopen('c:'dolzhenkov.txt',WRITE):
> fprintf(file_,convert(pl,string)):
> fclose(file_):
```

Отображение параболического множества и расчет каустики

```
> restart:with(Student[Calculus1]):with(plots):
with(plottools):
```

Шаг для численного дифференцирования и число отрезков разбиения для численного интегрирования

```
> h:=0.001:int_part_count:=60:
```

Параметр численного решения дифференциальных уравнений опытно выявлен как необходимый (иногда) к увеличению от значения по умолчанию

```
> AErr:=Float(1,-7):
```

Базис пространства

```
> e:=evalf([seq(sqrt(2)*sin(Pi*k*t),k = 1 .. 10)]);
> lambda:=evalf(4*Pi2/2+3);
```

Задание оператора уравнения. Здесь и далее процедурная обертка (оформление через proc) используется потому, что результат численного решения дифференциальных уравнений имеет тип proc, и для того, чтобы его можно было складывать с другими функциями или осуществлять композицию, надо, чтобы все операнды имели такой же тип.

Нелинейная часть оператора уравнения

```
> fn:=x->lambda*(sin(x)+1/2*sin(2*x)):
```

Процедурная обертка нелинейной части

```
> fnp:=proc(x)proc(s)eval(fn(x),t=s);end;end;
```

Оператор уравнения и интегранд его потенциала

```
> f:=x->diff(x,t$2)+fn(x):
> iV:=(x,q)->diff(x,t)2/2+lambda*(cos(x)-
-1)+lambda*1/4*(cos(2*x)-1)+(q)*x:
```

Далее следующим образом осуществляется переход к операторному уравнению с сжимающим оператором.

```
> Diff(x,t$2)+'f'(x)=0,G=(Diff(x,t$2))('-1');
Diff(x,t$2)=y,x=G(y); y+'f'(G(y))=0;
```

Поскольку решение уравнения ищется не на всем пространстве, то реально мы имеем дело со следующими представлениями и решением следующего уравнения:

```
> x=u+v,u=xi[1]*'e'[1]+xi[2]*'e'[2];y=y(u)+
+y(v),y=-Pi2*xi[1]*'e'[1]-4*Pi2*xi[2]*'e'[2]+
+y(v);
> y(v)+P[E[infinity-'2']]*'f'(u+G(y(v)))=0;
```

Численная реализация (как решение дифференциального уравнения) оператора обращения двойного дифференцирования с краевыми условиями

```
> G:=y->rhs(dsolve(diff(x(t),t$2)=y,x(0)=0,
x(1)=0,x(t),numeric, method=bvp,abserr=AErr,
maxmesh=256, output=listprocedure)[2]):
```

Процедурная реализация y(u)

```
> u:=proc(x,y)
> proc(s)eval(x*e[1]+y*e[2],t=s);end;
> end;
```

Процедурная обертка проекторов на
> e[1]; и
> e[2]; для процедурных функций (интегралы берутся по отрезку [0,1-h], поскольку вторая производная будет считаться через правую разность).

```
> projector1:=proc(x)
> evalf(ApproximateInt(x(t)*e[1],t=0..1-h,
method = trapezoid, partition=int_part_count));
> end:
> projector2:=proc(x)
> evalf(ApproximateInt(x(t)*e[2],t=0..1-h,
method = trapezoid, partition=int_part_count));
> end:
```

Процедурная обертка проектора на
> E[infinity-‘2‘]; для процедурных функций

```
> proj:=proc(x)
> local p1,p2:
> p1:=projector1(x):
> p2:=projector2(x):
> x-proc(s)eval(p1*e[1]+p2*e[2],t=s);end;
> end:
```

Начальный шаг для всех итерационных процессов в дальнейшем и тест правильности работы последнего проектора

```
> start:=proc(s)eval(e[3],t=s);end;
> proj(start)(1/2);
```

Процедурная реализация итерационного процесса для уравнения

```
> y(v)+P[E[infinity-‘2‘]]*f(G(y)) = 0;.
> Iter:=proc(x,y,st)
> local it:
> it:=u(x,y)+G(st(t)):
> proj(-fnp(it(t)));
> end;
```

Реализация нескольких итераций и получение v (x=u+v).

```
> Reduction:=proc(x,y,st,itcount)
> local z,i:
> z:=st:
> for i from 1 to itcount do
> z:=Iter(x,y,z):
> od:
> G(z(t));
> end;
```

Слагаемое u представления x=u+v

```
> _u:=proc(x,y)
> proc(s)eval(evalf(x)*e[1]+evalf(y)*e[2],t=s);
end;
> end;
```

Вычисление каустики в точке

```
> Kaust:=proc(x,y)
> local Red,dx,dy,f,diffp:
```

Вычисление производных ключевой функции как проекций градиента.

1. Вычисление v

```
> Red:=Reduction(x,y,start,2):
```

2. Вычисление градиента без учета

```
> Diff(u,‘$‘(t,2));.
> f:=fnp(_u(x,y)(t)+Red(t)):
```

3. Вычислений проекций и учет

```
> Diff(u,‘$‘(t,2));.
> dx:=projector1(f)-evalf(Pi2)*x:
> dy:=projector2(f)-evalf(4*Pi2)*y:
> [dx,dy];
> end;
> pl1:=op(1,pl):nops(pl1);
> for i from 1 to nops(pl1)-1 do
> s:=op(i,pl1);
> p1:=s[1]:
> pp1:=Kaust(p1[1],p1[2]):
> p1:=s[2]:
> pp2:=Kaust(p1[1],p1[2]):
> s:=[pp1,pp2]:
> if (i mod 20=0) then print(i,s);fi;
> pl1:=subsop(i=s,pl1):
> od:pl1:=subsop(1=pl1,pl):
> pl1;
> pl:=display(pl1,pl2):
> pl2:=reflect(pl,[[0,0],[0,1]]):
pl3:=display(pl,pl2):pl4:=reflect(pl3,[[0,0],[1,0]])
:pl5:=display(pl3,pl4):pl5;
> pl6:=display(pl5,view=[0..56,0..170]):pl6;
> A:=[[1,0],[13,4],[0.2,21.3],[8,39],[16,24],
[37,72]]:
> B:=[“A”,”B”,”C”,”D”,”E”,”F”,”G”,”H”,”I”,”K”]:
> Len:=[10,10,10,10,10,10,35,0.7,1.3,1.5,1.5]:
> pl_1:=seq(line(A[i],[A[i][1]+Len[i]*cos(Pi/4),
A[i][2]+Len[i]*sin(Pi/4)]),i=1..nops(A)):
> pl_2:=seq(textplot([A[i][1]+(Len[i])*cos(Pi/4)+
+1.5,
A[i][2]+(Len[i])*sin(Pi/4)+1.5,B[i]],font=
=[TIMES, ROMAN,20]),i=1..nops(A)):
> display(pl6,pl_1,pl_2,titlefont=[TIMES,ROMAN,
25]);pl7:=
> display(pl5,view=[0..1.1,181..196]);pl8:=
> A:=[[0.14,182.8],[0.02,183.5],[0.14,187]]:
```

```
> B:=["G","H","I","K"]:
> Len:=[0.2,0.45,0.2,10,10,10,35,0.7,1.3,1.5,1.5]:
> pl_1:=seq(line(A[i],[A[i][1]+Len[i]*cos(Pi/4),
A[i][2]+Len[i]*sin(Pi/4)]),i=1..nops(A)):
> pl_2:=seq(textplot([A[i][1]+(Len[i])*
*cos(Pi/4)+0.025,
A[i][2]+(Len[i])*sin(Pi/4)+0.025,B[i]], font=
=[TIMES,ROMAN,20]),i=1..nops(A)):
> display(pl8,pl_1,pl_2,titlefont=[TIMES,ROM
AN,25]);pl9:=
> pl:=display(pl1,pl2):pl;
> pl2:=reflect(pl,[[0,0],[0,1]]):pl3:=display(pl,pl2):
pl4:=reflect(pl3,[[0,0],[1,0]]):pl5:=display(pl3,
pl4):pl5;
```

5. РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ

Приведем результаты вычислений по представленным выше программам.

Для схемы Ляпунова—Шмидта: экстремумы ключевой функции являются отрезками ряда Фурье — проекциями решения задачи на линейную оболочку ключевых параметров. Если зафиксировать теперь экстремум ключевой функции и редуцироваться к ключевой функции на единицу большего числа переменных, то первоначальная функция одной переменной будет уже, очевидно, иметь единственный экстремум, отыскав который, зафиксируем его и увеличим еще на единицу размерность пространства определения ключевой функции. Проделаем это то количество раз, которое обеспечит нам приемлемую точность.

Так, для уравнения 2-SG при $\lambda = 2\pi^2 + 3$ если мы зафиксируем седловую критическую точку (мы взяли $q_1 = 8, q_2 = 39$), этот процесс дает нам последовательность приближений коэффициентов Фурье, по шести членам которой получаем рис. 5.

Аналогично для правого минимума $\lambda = 2\pi^2 + 3$, ($q_1 = 0.02, q_2 = 183.5$).

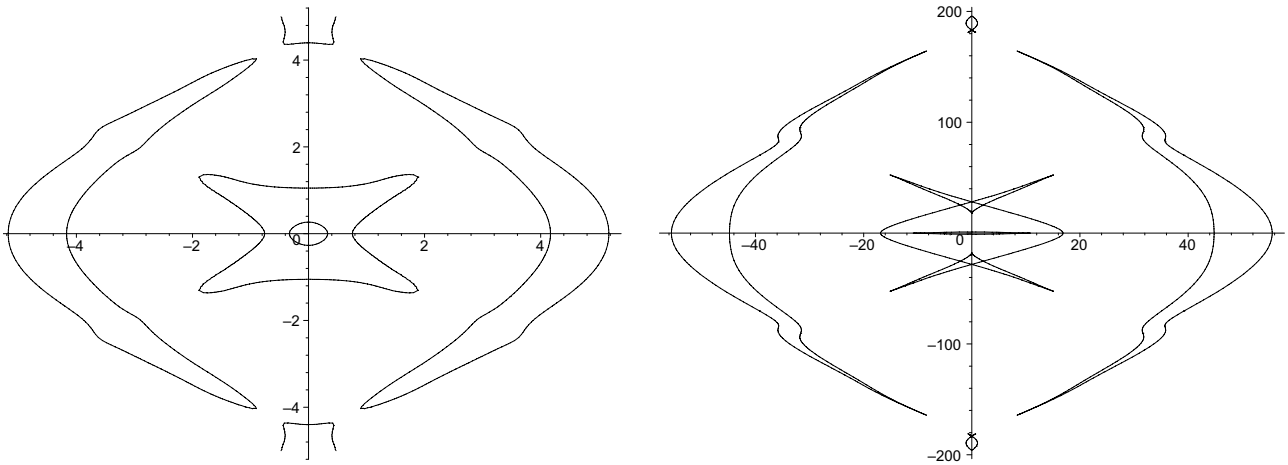

Рис. 1. Фрагменты параболического множества и каустики приближения ключевой функции при $\lambda = (4\pi^2)/2 + 3$

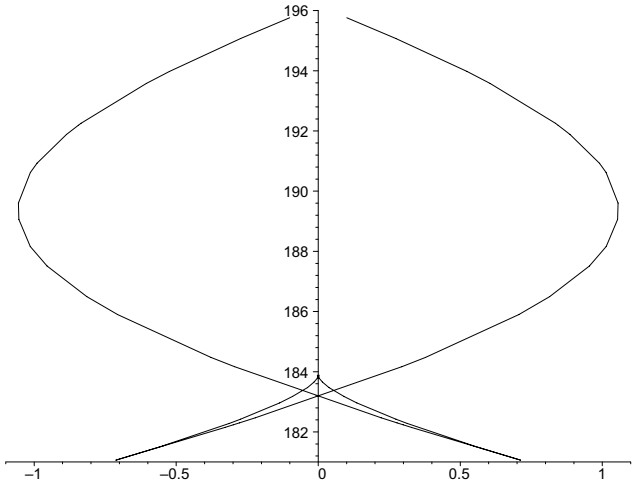

Рис. 2. Фрагмент каустики в ином масштабе

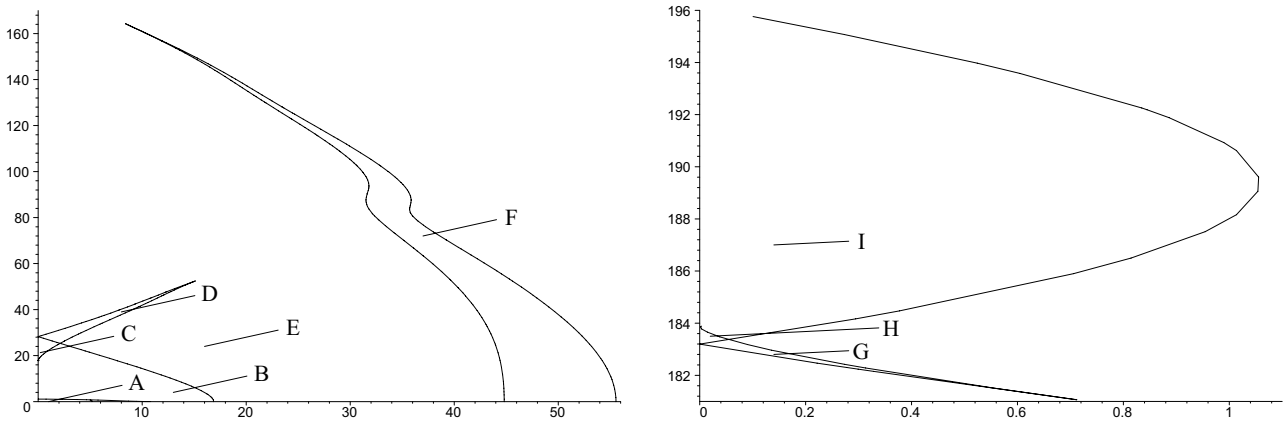

Рис. 3. Выделенные области фрагментов каустики

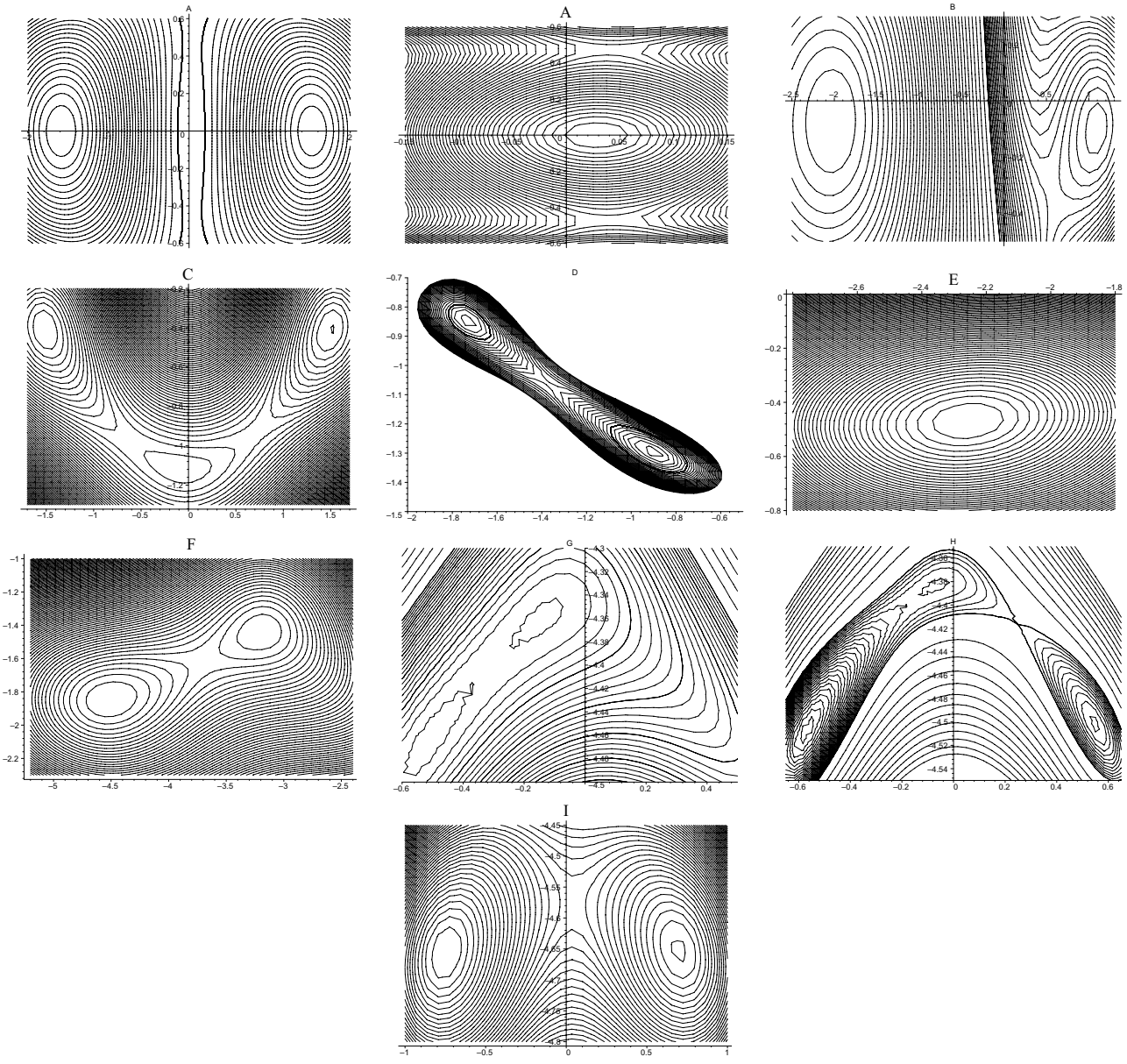

Рис. 4. Линии уровня ключевой функции для выделенных областей параметров (для первой области — в двух масштабах)

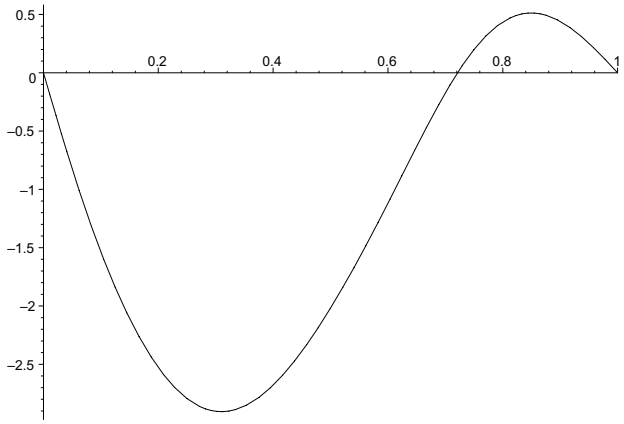

Рис. 5. Приближение указанного решения

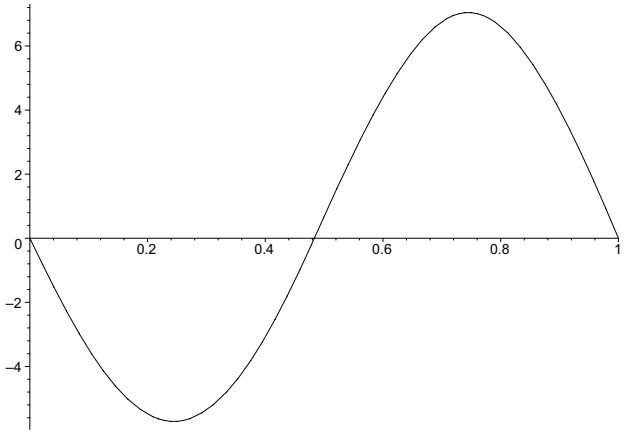

Рис. 6. Приближение указанного решения

ЛИТЕРАТУРА

1. *Борисович Ю. Г.* Нелинейные фредгольмовы отображения и теория Лере—Шаудера / Ю. Г. Борисович, В. Г. Звягин, Ю. И. Сапронов // Успехи матем. наук. — 1977. — Т. 32, Вып. 4. — С. 3—54.

2. *Инфельд Э.* Нелинейные волны, солитоны и хаос / Э. Инфельд, Дж. Роуландс. —М. : ФИЗМАТЛИТ, 2006. — 480 с.

3. *Красносельский М. А.* Об одной схеме исследования вырожденных экстремалей функционалов классического вариационного исчисления / М. А. Красносельский, Н. А. Бобылев, Э. М. Мухамадиев // ДАН СССР. — 1978. — Т. 240, № 3. — С. 530—533.

4. *Бобылев Н. А.* Геометрические методы в вариационных задачах / Н. А. Бобылев, С. В. Емельянов, С. К. Коровин. — М. : Магистр, 1998. —658 с.

5. *Marsden J. E.* On the geometry of the Liapunov-Schmidt procedure / J. E. Marsden // Lect. Notes in Math. — 1979. — Vol. 755. — P. 77—82.

6. *Сапронов Ю. И.* Конечномерные редукции в гладких экстремальных задачах / Ю. И. Сапронов // Успехи матем. наук. — 1996. — Т. 51, Вып. 1. — С. 101—132.

7. *Даринский Б. М.* Бифуркации экстремалей фредгольмовых функционалов / Б. М. Даринский, Ю. И. Сапронов, С. Л. Царев// Современная математика. Фундаментальные направления. — Т. 12 (2004). — М., 2004. — С. 3—140.