

## ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА РЕШЕНИЯ ДВУМЕРНЫХ И ТРЕХМЕРНЫХ ЗАДАЧ ТЕОРИИ УПРУГОСТИ НА БАЗЕ MAPLE

В статье приведено описание инструментальной системы решения двумерных и трехмерных задач теории упругости. Вычислительное ядро системы составляет Maple. Произведена формализация основных алгоритмов с целью нахождения операций программирования указанных задач. В качестве примеров работы с системой приведены решения конкретных задач.

**Ключевые слова:** препроцессор, процессор, постпроцессор, метод начальных функций.

### ВВЕДЕНИЕ

В настоящее время не теряют своей актуальности аналитические методы, которые позволяют получать точные и обобщенные результаты. Благодаря развитию систем компьютерной математики (СКМ) стала возможной реализация операций аналитического дифференцирования и интегрирования. Это открывает большие возможности в применении СКМ для математического моделирования. Однако системы не обладают достаточно развитыми способами решения технических задач. Большинство из них ориентировано на применение в алгебре и математическом анализе [1]. Поэтому в процессе математического моделирования их необходимо дорабатывать, чтобы иметь возможность применить для решения той или иной сложной технической задачи. Создав препроцессор в СКМ и воспользовавшись вычислительными функциями СКМ, можно построить средство для решения конкретных задач.

В предлагаемой работе дана характеристика одного из средств для решения статических задач линейной теории упругости – инструментальной системы, основанной на базе ядра СКМ Maple. Система Maple была выбрана авторами как система, наиболее подходящая по входному языку. С помощью программирования в ее среде можно создать эффективное вычислительное средство для решения задач теории упругости. В разработанной инструментальной системе решаются задачи теории плит, пластин, полуплоскости и полупространства. Производятся сложные аналитические операции над формульным заданием выражений теории упругости. Вопросами разработки инструментальных систем для решения задач теории упругости с внедрением аналитических методов занимались Кабулов В. К., Толоч В. О. [2]; построением инструментальной системы на ос-

нове метода конечных элементов занимался Гоменюк С. И. [3]. Их работы способствовали разработке новой системы на основе Maple.

### ОПИСАНИЕ СИСТЕМЫ

Входной язык – расширенный язык Maple. Алфавит языка Maple включает 26 букв латинского алфавита, строчных (a–z) и прописных (A–Z), десять цифр (0–9) и 32 специальных символа. Подробное описание языка приведено в работе [4]. Расширения языка процедуры и функции: `sdiff` – символическое дифференцирование; `sint` – символическое интегрирование; `operators` – процедура вывода операторов решения плиты, пластины; `mch` – процедура разложения гиперболического косинуса в ряд по Маклорену; `msh` – разложение гиперболического синуса; `p` – перевод символических рядов в числовые для двумерной задачи; `nm` – перевод в числовые ряды для трехмерного случая; `integral` – процедура интегрирования по параметру; `galerkin2D` – процедура Галеркина для двумерной задачи; `galerkin3D` – трехмерной; `maklogen` – превращение тригонометрических функций в разложения Маклорена; `symbolism` – проведение символических операций дифференцирования и интегрирования; `semi` – процедура решения для полуплоскости и полупространства; `seminitial` – процедура начальной инициализации, задание фундаментального решения.

Предлагаемая система состоит из трех основных блоков: препроцессора, процессора и постпроцессора. Препроцессор – набор процедур, предназначенный для автоматизации подготовки исходных данных. В системе он занимается выводом решения в операторно-символической форме и заданием начальных функций и граничных условий задачи.

Препроцессор отвечает за:

- а) выбор размерности задачи (двумерная или трехмерная);
- б) выбор системы координат (декартова, полярная, цилиндрическая);
- в) реализация символьных операций дифференцирования и интегрирования;
- г) вывод решения в символьной форме;
- д) вывод решения в виде операторов [5].

Препроцессор занимается обработкой вводимых пользователем данных. Перед решением задачи на языке инструментальной системы необходимо описать входные и выходные параметры, далее с помощью подключения библиотеки осуществить процесс решения поставленной задачи.

Процессор – важная часть работы системы. Процесс его работы можно условно разбить на следующие этапы:

- а) перевод решения из символьной формы в интегрально-дифференциальную;
- б) проведение аналитического дифференцирования, интегрирования [6];
- в) вывод аналитического решения задачи.

В роли постпроцессора выступают функции и процедуры, осуществляющие числовой счет по выведенной аналитической формуле. Далее графические процедуры системы визуализируют результат.

Автоматизация аналитического вывода решения и последующий его перевод с символьного представления в формульный вид представляет собой отдельную сложную задачу, которая была осуществлена путем создания библиотеки подпрограмм и функций для Maple. В библиотеке содержатся команды препроцессора и процессора.

### Строение препроцессора

**Алгоритм 1.** В препроцессор системы вошел алгоритм построения аналитического решения для балок, плит и пластин в двумерной и трехмерной постановке. Суть его в следующем: решается система из четырех уравнений теории упругости для двумерной постановки и из шести для трехмерной [7], [8].

Используется решение бигармонического уравнения  $\nabla\nabla U = 0$  в форме операторно-символьных  $\Phi$ -функций. Процесс вывода детально описывается в работе [5]:

$$f(x, y, z) = f_0(x, y)\cos(\gamma z) + f_1(x, y)\sin(\gamma z) + f_2(x, y)z\cos(\gamma z) + f_3(x, y)z\sin(\gamma z), \quad (1)$$

где  $\gamma = \sqrt{\frac{d^2}{dx^2} + \frac{d^2}{dy^2}} = \sqrt{\alpha^2 + \beta^2}$  – сумма производных  $\alpha = \frac{d}{dx}$ ,  $\beta = \frac{d}{dy}$ ;  $\sin$ ,  $\cos$  – символические разложения Маклорена.

В случае двумерной задачи  $\Phi$ -функция записывается в виде

$$f(x, y) = f_0(x)\cos(\alpha y) + f_1(x)\sin(\alpha y) + f_2(x)y\cos(\alpha x) + f_3(x)y\sin(\alpha y),$$

где  $\alpha = \frac{d}{dx}$  – символическое обозначение производной.

Далее программируется второй метод из работы [5]. Однако, чтобы проводить аналитические операции над символьными разложениями Маклорена  $\sin$  и  $\cos$ , как над обычными функциями, требовалось доказать, что эти разложения обладают свойствами тригонометрических функций. Именно этому вопросу и была посвящена работа [9]. Благодаря доказательству закона ортогональности для общей трехмерной задачи все символьные преобразования над разложениями Маклорена  $\sin$  и  $\cos$  в системе считаются допустимыми и дают правильные результаты. По методу  $\Phi$ -функций решаются уравнения теории упругости [5], [7], [8].

Этот процесс реализован в процедуре operators.

```
> S1:= eval(value(S), z = 0);
S2:= eval(diff(value(S), z), z = 0);
> S3:= eval(diff(value(S), `$(z, 2)), z = 0);
[S1[], S2[], S3[]];
solve(% , [u1, u2, u3, v1, v2, v3, w1, w2, w3, xx1, xx2, xx3,
yy1, yy2, yy3, zz1, zz2, zz3]); assign(%); factor(%);
> R:= collect(simplify(algsubs(alpha^2+beta^2 = gamma^2,
%)), gamma, factor)[];
subs(R, U); simplify(%); collect(% , u0); collect(% , v0);
collect(% , w0); collect(% , zz0); collect(% , yy0);
U:= sort(collect(% , xx0), [u0, v0, w0, zz0, yy0, xx0]);
subs(R, V); simplify(%); collect(% , u0); collect(% , v0);
collect(% , w0); collect(% , zz0); collect(% , yy0);
V:= sort(collect(% , xx0), [u0, v0, w0, zz0, yy0, xx0]);
subs(R, W); simplify(%); collect(% , u0); collect(% , v0);
collect(% , w0); collect(% , zz0); collect(% , yy0);
W:= sort(collect(% , xx0), [u0, v0, w0, zz0, yy0, xx0]);
subs(R, Z); simplify(%); collect(% , u0); collect(% , v0);
collect(% , w0); collect(% , zz0); collect(% , yy0);
Z:= sort(collect(% , xx0), [u0, v0, w0, zz0, yy0, xx0]);
subs(R, Y); simplify(%); collect(% , u0); collect(% , v0);
collect(% , w0); collect(% , zz0); collect(% , yy0);
Y:= sort(collect(% , xx0), [u0, v0, w0, zz0, yy0, xx0]);
subs(R, X); simplify(%); collect(% , u0); collect(% , v0);
collect(% , w0); collect(% , zz0); collect(% , yy0);
X:= sort(collect(% , xx0), [u0, v0, w0, zz0, yy0, xx0]);
```

В процедуре используются следующие операторы и функции языка Maple. Функция eval(<expr>, <пере-

менная = значение>) – вычисление выражения при заданном значении входящей в него переменной; <expr> – выражение, которое необходимо вычислить, <переменная> – имя входящей в <expr> переменной. Функция value(<expr>) – вычисление выражения, в которое входят инертные переменные. Оператор diff(<expr>,<переменная>\${порядок производной}) – взятие производной в аналитической форме. Функция solve(<уравнение|[система уравнений]>{система уравнений}>,<решение|[решения]>{решения}>) – аналитическое решение уравнения или системы уравнений. При этом система уравнений может задаваться в виде списка перечисляемых уравнений, обладающего упорядоченностью, или в виде множества уравнений. Порядок следования уравнений во множестве не важен. Список уравнений заключается в прямые скобки [], множество – в фигурные {}. После вывода решений переменным необходимо присвоить их значения, для этого используется функция assign(<expr>). Присвоенные значения переменных преобразовываются с помощью функции factor(%) – разложение полинома на множители. Функция algsubs(<переменная1 = значение1>, <переменная2 = значение2>, ..., <переменнаяN = значениеN>, <expr>) – алгебраическая замена переменных их значениями и алгебраическое преобразование выражения <expr>. Функция subs(<переменная1 = значение1>, <переменная2 = значение2>, ..., <переменнаяN = значениеN>, <expr>) – замена переменных их значениями без алгебраического преобразование выражения. Функция simplify(<expr>|<методы упрощения>|<assume = свойство>) – упрощение математического выражения <expr> по заданным методам <методы упрощения>. В качестве методов могут быть выбраны опции exp, ln, sqrt, trig, radical, power, свойство assume – указание диапазона значений, принимаемых переменными в упрощении, может принимать значения complex – комплексные значения, real – действительные числа, positive – положительные числа, integer – целые числа, RealRange(a, b) – интервал действительных чисел (a, b). Функция collect(<expr>, <набор переменных>|<методы преобразования>) – функция приведения подобных членов выражения <expr> по выбираемым именам переменных <набор переменных>; <методы преобразования> – перечисление методов преобразований выражений за скобками приведенных членов. Может принимать значения combine – представление в форме произведения скобок, factor – разложение на множители, normal – сокращение дробей, simplify – полное упрощение, recursive – приведение подобных членов при степенях первой переменной в списке, далее по второй и т. д., distributed – при-

ведение коэффициентов при членах, содержащих всевозможные произведения степеней переменных в списке или множестве, причем суммарная степень всех переменных возрастает от наименьшей к наибольшей. По умолчанию в collect работает метод recursive. Функция sort(<expr>) – сортировка полинома по степеням от наибольшей к наименьшей.

Получается общее решение, из которого выделяются операторы [9]. Этот процесс осуществляет процедура operat.

```
> operat:=proc(res, per, vec)
local oper, dn, fr;
if has(oper, sin) then
oper:=simplify(eval(subs(vec, res)/per,
gamma=sqrt(alpha^2+beta^2)));
oper:=collect(simplify(subs(alpha^2+beta^2=gamma^2,
oper)),[gamma, cos, sin, nu]);
dn:=simplify(numer(oper)); fr:=denom(oper);
dn:=collect(dn,[gamma, cos, sin, nu, z]);
fr:=collect(fr, gamma);
dn:=subs(alpha^2+beta^2=gamma^2, dn);
dn/fr;
else
oper:=simplify(eval(subs(vec, res)/per,
gamma=sqrt(alpha^2+beta^2)), power, radical, sqrt);
oper:=combine(simplify(subs(alpha^2+beta^2=gamma^2,
oper), power, radical, sqrt));
dn:=numer(oper); fr:=denom(oper);
dn:=collect(dn,[per, nu, gamma, z, exp]);
fr:=collect(fr, gamma);
dn:=subs(alpha^2+beta^2=gamma^2, dn);
dn/fr; fi;
end proc;
```

В указанных выше процедурах добавлены процедуры numer(<expr>) – выделение числителя выражения <expr> и denom(<expr>) – знаменателя, соответственно, и оператор has(<expr>,<подвыражение>), который возвращает логическое значение true, если <подвыражение> содержится в <expr>.

**Алгоритм 2.** В препроцессор был введен алгоритм, позволяющий получать решения задач теории упругости для полуплоскости. Основная идея алгоритма принадлежит Толоку В. О., аспирантом Овским А. Г. он был реализован в Maple.

Используются функции библиотеки Vlasov. m. Основная идея алгоритма следующая.

Решение бигармонического уравнения  $\nabla\nabla W = 0$  строится в виде  $w(x, y) = w_0(x, y) + u\psi(x, y)$ , где  $w_0(x, y)$  и  $\psi(x, y)$  – гармонические функции-решения уравнения  $\Delta_2 w = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}$ . Общее решение краевой задачи в случае полуплоскости для уравнения Лапласа  $\Delta_2 w = 0$  имеет вид  $w_0(x, y) =$

$= \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y w_0(\xi) d\xi}{(x-\xi)^2 + y^2}$ . Для начальной прямой решение

бигармонического уравнения будет иметь вид

$$\frac{\partial w}{\partial y} \Big|_{y=0} = \frac{\partial w_0}{\partial y} + \psi + y \frac{\partial \psi}{\partial y} \Big|_{y=0};$$

$$w_1(x, y) = \frac{\partial w_0}{\partial y} + \psi, \quad (2)$$

где  $w_1 = w_1(x, y) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y w_1(\xi) d\xi}{(x-\xi)^2 + y^2}$ ,  $w_1(\xi)$  – находится из уравнений теории упругости в процессе решения задач [10]. Выразим гармоническую составляющую  $\psi$ . Решив уравнение, получим:

$$w(x, y) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y^3 w_0(\xi) d\xi}{((x-\xi)^2 + y^2)^2} + \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y w_1(\xi) d\xi}{(x-\xi)^2 + y^2}. \quad (3)$$

Запишем выражение (3) для  $\sigma_y(x, y)$ , имеем

$$\sigma_y(x, y) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y^3 w_y^0(\xi)}{((x-\xi)^2 + y^2)^2} d\xi + \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y^2 w_y^1(\xi)}{((x-\xi)^2 + y^2)} d\xi, \quad (4)$$

где  $\sigma_y^1 = \frac{\partial \sigma_y^0}{\partial y} = \frac{\partial \tau_0}{\partial x}$  – выражение для производной напряжения  $\sigma_y$  на прямой  $y = 0$ , согласно Власову [7].

Произведя все подстановки и учитывая свойство подынтегральных выражений, получим:

$$\sigma_y(x, y) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y^3}{r^4} \sigma_y^0(\xi) d\xi + \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2(x-\xi)y^2}{r^4} \tau_{xy}^0(\xi) d\xi. \quad (5)$$

Здесь  $r^4 = ((x-\xi)^2 + y^2)^2$  – сокращенное обозначение. Далее подставляя в уравнения теории упругости для двумерного случая [8] и произведя интегрирования, получим выражения для перемещений и напряжений в случае плоской деформации:

$$\tau_{xy}(x, y) = \int_0^{x-\xi} \frac{\partial \sigma_y}{\partial y} dx = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y^2(x-\xi)}{r^4} \sigma_y^0(\xi) d\xi + \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y(x-\xi)^2}{r^4} \tau_{xy}^0(\xi) d\xi;$$

$$\sigma_x = - \int_0^{x-\xi} \frac{\partial \sigma_{xy}}{\partial y} dx = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y(x-\xi)^2}{r^4} \sigma_y^0(\xi) d\xi + \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2(x-\xi)^3}{r^4} \tau_{xy}^0(\xi) d\xi;$$

$$u = \int_0^{x-\xi} \frac{1}{2G} [(1-\nu)\sigma_x - \nu\sigma_y] dx = \frac{1}{2G\pi} \int_{-\infty}^{\infty} \left[ (1-2\nu) \arctg \frac{x-\xi}{y} - \frac{(x-\xi)y}{r^2} \right] \sigma_y^0(\xi) d\xi + \frac{1}{2G\pi} \int_{-\infty}^{\infty} \left[ (1-\nu)(\ln r^2 + C) - \frac{y^2}{r^2} \right] \tau_{xy}^0(\xi) d\xi;$$

$$v = \int_0^{x-\xi} \left( \frac{\tau_{xy}}{G} - \frac{\partial u}{\partial y} \right) dx = \frac{1}{2G\pi} \int_{-\infty}^{\infty} \left[ (1-\nu)(\ln r^2 + C) - \frac{y^2}{r^2} \right] \sigma_y^0(\xi) d\xi - \frac{1}{2G\pi} \int_{-\infty}^{\infty} \left[ (1-2\nu) \arctg \left( \frac{x-\xi}{y} \right) - \frac{(x-\xi)y}{r^2} \right] \tau_{xy}^0(\xi) d\xi. \quad (6)$$

Преобразования (5), (6) в разработанной системе реализуются с помощью следующих алгебраических операций:

$$\sigma_y := \phi \sigma_0 + \psi \tau_0;$$

$$\tau := - \frac{\beta \phi \sigma_0}{\alpha} - \frac{\beta \psi \tau_0}{\alpha};$$

$$\sigma_x := \frac{\beta^2 \phi \sigma_0}{\alpha^2} + \frac{\beta^2 \psi \tau_0}{\alpha^2};$$

$$\sigma_y := - \alpha \phi u_0 - \alpha \psi u_1 - \frac{\alpha^2 \phi v_0}{\beta} - \frac{\alpha^2 \psi v_1}{\beta};$$

$$u := \frac{1}{2} \frac{\beta^2 \phi \sigma_0}{\alpha^3 G} + \frac{1}{2} \frac{\beta^2 \psi \tau_0}{\alpha^3 G} - \frac{1}{2} \frac{\beta^2 \nu \phi \sigma_0}{\alpha^3 G} - \frac{1}{2} \frac{\beta^2 \nu \psi \tau_0}{\alpha^3 G} - \frac{1}{2} \frac{\nu \psi \tau_0}{\alpha G} - \frac{1}{2} \frac{\nu \phi \sigma_0}{\alpha G};$$

$$v := \frac{1}{2} \frac{\beta \nu \phi \sigma_0}{\alpha^2 G} + \frac{1}{2} \frac{\beta \nu \psi \tau_0}{\alpha^2 G} - \frac{1}{2} \frac{\nu \psi \tau_0}{\beta G} - \frac{1}{2} \frac{\nu \psi \tau_0}{\beta G} - \frac{1}{2} \frac{\nu \phi \sigma_0}{\beta G} + \frac{1}{2} \frac{\phi \sigma_0}{\beta G}.$$

Здесь  $\alpha = \frac{d}{dx}$ ,  $\beta = \frac{d}{dy}$ ,  $\frac{1}{\alpha} f = \int_0^{x-\xi} f dx$ ,  $\frac{1}{\beta} f = \int_0^y f dy$  – замена операций интегрирования и дифференцирования на умножение и деление;  $\phi = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{2y^3}{((x-\xi)^2 + y^2)^2} d\xi$ ,  $\psi = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y^2}{((x-\xi)^2 + y^2)} d\xi$  – сокращенные обозначения;  $G$  – модуль сдвига.

Преобразования в препроцессоре осуществляются в строгом порядке без нарушения преобразований.

Все операции дифференцирования и интегрирования сведены к умножению и делению.

### Строение процессора

После предварительной подготовки и указания граничных условий начинает работать процессор. Он переводит операторно-символическую форму решения в формульную. Осуществляется для каждого из полученных операторов с помощью процедуры nm для трехмерного и n для двумерного случая. Учитывается свойство тригонометрических функций в рядах (2) и (3) менять знак при дифференцировании. После взятия всех производных разложения Маклорена в символическом решении результат преобразуется в числовой [8].

Текст на языке Maple процедур nm и n:

```
> vlasov[nm]:=proc(res, per)
local i:: integer, rs, op1, op2, op3, po:: anything, str:: string;
global z, alpha, beta, gamma;
op1:=op(per);
op2:=op(op1[1]);
op3:=op(op2[1]);
rs:= nops(op2[1]); po:=1;
for i from 1 to rs do
po:=po*op3[i]; od;
rs:=res;
if po<>0 then
rs:=subs(sin=sinh, cos=cosh, rs);
rs:=subs(alpha^3=diff(po, x$3)/po, beta^3=diff(po, y$3)/
po, rs);
rs:=subs(alpha^2=diff(po, x$2)/po, beta^2=diff(po, y$2)/
po, rs);
rs:=subs(alpha=diff(po, x)/po, beta=diff(po, y)/po, rs); fi;
rs*po;
end proc;
> vlasov[n]:=proc(res, per)
local i:: integer, rs, po:: anything, str:: string;
global y, alpha, gamma;
po:=op(per)[1];
rs:=res;
if po<>0 then
rs:=subs(sin(alpha*y)=sinh(gamma*y),
cos(alpha*y)=cosh(gamma*y), rs);
rs:=expand(rs);
rs:=subs(alpha^3=diff(po, x$3)/po, rs);
rs:=subs(alpha^2=diff(po, x$2)/po, rs);
rs:=subs(alpha=diff(po, x)/po, rs); fi;
rs*po;
end proc;
```

После этого применяется процедура Галеркина и получается результат в числовой форме:

```
> vlasov[galerkin2D]:= proc (inp, r:: anything)
local st; global gamma, y, S;
integral(inp*r, x = 0.. S)
end proc;
> vlasov[galerkin3D]:= proc (inp, r1, r2:: anything)
local str, t;
```

```
global gamma, z, Q, S, R;
str:= inp;
t:= remove(has, str, r1);
str:= select(has, str, r2);
str:= simplify(select(has, str, r1)/(r1*r2));
(1/4)*str*t*S*R end proc;
```

В процедурах используются функции работы со структурой выражений `op(<expr>)` – подъем по бинарному дереву структуры выражения вверх, в результате получается последовательность выражений. Функция `nops(<expr>)` – подсчет количества выражений в последовательности выражений `op`. Функция `expand(<expr>)` раскрывает скобки в выражении `<expr>`. Используются также процедуры выделения подвыражения из заданного выражения `select(has, <expr>, <выражение>)`, где `has` указывает на выделение подвыражения, где встречается выражение имени, или оператора. Функция `remove(has, <expr>, <выражение>)`, наоборот, удаляет выделяемое подвыражение.

В библиотеку вошли созданные автором процедура интегрирования и процедура взятия модуля числа. Стандартные процедуры системы Maple не рассчитаны на обработку сложных аналитических выражений. Этот недостаток Maple вызывал зависание вычислительного ядра инструментальной системы при расчете задач методом начальных функций [6]. Он был программным способом устранен автором. Ниже приведен текст на языке Maple процедур интегрирования и взятия модуля числа.

```
> vlasov[fabs]:=proc(inp)
local per;
per:=evalf(inp);
if per<0 then per:=-per; fi;
per;
end proc;
> vlasov[integral]:=proc(input, perem)
local arg, gran1, gran2, result;
arg:=op(perem)[1]; op(rhs(perem));
gran1:=%[1]; gran2:=%[2];
result:=int(input, arg);
simplify(eval(result, arg=gran2)-eval(result, arg=gran1));
end proc;
```

Для того чтобы решить задачи в системе, необходимо запустить среду Maple и подключить откомпилированную библиотеку к создаваемому документу. Для этого в режиме `worksheet` Maple набирается следующая последовательность команд [5]:

```
> restart;
> read(`c:/Vlasov. m`);
> with(vlasov);
```

### Решение задач теории плит, пластин

Для решения задач теории упругости для плит, пластин необходимо вызвать процедуру `operators`

со следующим списком параметров в скобках: *vaxes* – обязательный параметр, задает систему координат (*cartesian* – обычная декартова система, *polar* – полярная); *vdimension* – обязательный параметр, задающий размерность задачи (2 – плоская задача, 3 – трехмерная); *vchange* – необязательный параметр, он меняет ось, относительно которой строится решение в плоской задаче (принимает значение *yes*). По умолчанию строится решение относительно оси *y*, на практике нужно получать решение и для оси *x*; *vtype* – тип решения, *bigarmonic* – бигармоническое (*exp* – экспоненциальное). Для трехмерной задачи вместо необязательного параметра *vchange* используется обязательный *vaxe* (*x*, *y*, *z* – оси вывода). После вызова процедуры *operators* и задания начальной инициализации автоматически получаем решение общей задачи теории упругости. Пример вызова процедуры указан ниже:

```
> operators(vaxes=cartesian, vdimension=3, vaxe=z,
vtype=bigarmonic):
> operators(vaxes=cartesian, vdimension=2,
vtype=bigarmonic):
```

Общее решение представляется в операторно-символической форме и хранится в переменных *U*, *V*, *X*, *Y*, *sigma\_x*, *sigma\_y*, *tau* в случае двумерной задачи и в *U*, *V*, *W*, *X*, *Y*, *Z*, *sigma\_x*, *sigma\_y*, *tau* – в трехмерном случае. Операторно-символическая форма, содержащая дифференциалы высших порядков, слишком сложна для проведения вычислительных операций. Поэтому, после подстановки начальных функций в общее дифференциальное решение двумерной и трехмерной задачи и проведения сложных математических операций, на выходе получается решение, содержащее числовые ряды [6]. Для двумерного случая эти операции осуществляются путем вызова процедур библиотеки *n(expr)*, где *expr* – решение в операторной форме (а именно, через операторы *Luu*, *Luv* и т. д.) и *nim(expr)* для трехмерной задачи. После перевода в числовую форму над решением производится процедура Галеркина. Для двумерной задачи это процедура *galerkin2D(expr, sin(a[n]\*x))*, трехмерной – *galerkin3D(expr, sin(a[n]\*x)\*cos(b[m]\*y))*. После этого строятся уравнения, с помощью которых решается задача и находятся коэффициенты *An*, *Bn* для двумерной задачи и *Anm*, *Bnm*, *Cnm* – для трехмерной.

#### Решение задач для полуплоскости, полупространства

Эти задачи в системе решаются следующим образом. Вызываются процедуры:

```
> semi(saxes = cartesian, sdimension = 2, stype = napr);
```

```
> phi :=  $\frac{2 \cdot y^3}{((x - xi)^2 + y^2)^2}$ ; psi :=  $\frac{y^2}{(x - xi)^2 + y^2}$ ;
R := sqrt((x - xi)^2 + y^2);
> seminital(phi, psi, R):
```

Для решения задачи используются две процедуры. Первая процедура *semi* имеет следующий список параметров в скобках: *saxes* – обязательный параметр, задает систему координат (*cartesian* – обычная декартова система, *polar* – полярная); *sdimension* – обязательный параметр, задающий размерность задачи (2 – плоская задача, 3 – трехмерная); *stype* – тип решения; *napr* – напряжения (*perem* – перемещения). Вызвав процедуру *semi*, выбираем тип задачи. Передав два параметра процедуре *seminital*, задаем, над какими функциями производятся операции. Третий параметр – переменная, относительно которой производятся упрощения. В поставленных задачах – это переменная *R*.

#### ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ ТЕОРИИ УПРУГОСТИ В ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЕ

При решении множества задач теории упругости для плит, оболочек, пластин их сводят с помощью различных инженерных гипотез и допущений к общим модельным задачам. Математическая модельная задача значительно проще при реализации расчета. Математическая модель вместе с математическим выражением инженерных гипотез составляет расчетную модель конструкции.

Расчетная модель дает математическую зависимость между основными компонентами теории упругости и новыми величинами, относительно которых будут построены все соотношения. Чаще всего в моделях используются перемещения, напряжения и величины, их заменяющие.

Рассмотрим детально процесс построения модели решения двумерных и трехмерных задач для плиты и массива.

**Задача 1.** Рассмотрим свободно опертую по краям плиту длины *S* и толщины *h*, изгибаемую нагрузкой постоянной интенсивности *P* [10].

Выбираются начальные функции в виде

$$U_0(x, y) = \sum_{n=1}^{\infty} \sin A_n \frac{n\pi}{S} x,$$

$$V_0(x, y) = \sum_{n=1}^{\infty} B_n \cos \frac{n\pi}{S} x.$$

На языке системы они записываются в виде:

```
> Uo:= Sum(Anm*sin(a[n]*x), n=1.. p);
> Vo:=Sum (Bnm*cos(a[n]*x), n=1.. p);
```

В результате коэффициенты  $A_n$  и  $B_n$  на Maple:

$$\begin{aligned}
 A_n = & - \left( 4h \sin(n\pi)^3 \left( e^{\frac{n\pi h}{S}} \right)^2 - 1 \right) P S^2 e^{\frac{n\pi h}{S}} (-1 + \nu) / \left( -2n^2 \pi^2 h^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^2 + n^2 \pi^2 h^2 \sin(n\pi)^4 + \right. \\
 & + n^2 \pi^2 h^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^4 + S^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^4 + S^2 \sin(n\pi)^4 - h^2 n^2 \pi^2 \cos(n\pi)^2 \sin(n\pi)^2 \left( e^{\frac{n\pi h}{S}} \right)^4 - \\
 & - 2h^2 n^2 \pi^2 \cos(n\pi)^2 \sin(n\pi)^2 \left( e^{\frac{n\pi h}{S}} \right)^2 + 2hn^2 \pi^2 \sin(n\pi) - 2hn^2 \pi^2 \left( e^{\frac{n\pi h}{S}} \right)^2 S \sin(n\pi)^2 - \\
 & \left. - h^2 n^2 \pi^2 \cos(n\pi)^2 \sin(n\pi)^2 - 2S^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^2 + h^2 n^4 \pi^4 + 2h^2 n^4 \pi^4 \left( e^{\frac{n\pi h}{S}} \right)^2 + h^2 n^4 \pi^4 \left( e^{\frac{n\pi h}{S}} \right)^2 \right); \\
 B_n = & 4 \sin n\pi \left( hn\pi \cos(n\pi) \sin(n\pi) \left( e^{\frac{n\pi h}{S}} \right)^2 + hn\pi \cos(n\pi) \sin(n\pi) - hn^2 \pi^2 \left( e^{\frac{n\pi h}{S}} \right)^2 - \right. \\
 & - hn^2 \pi^2 + S \sin(n\pi)^2 \left( e^{\frac{n\pi h}{S}} \right)^2 - S \sin(n\pi)^2 \left. \right) P S^2 e^{\frac{n\pi h}{S}} (-1 + \nu) / \left( n\pi \left( -2n^2 \pi^2 h^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^2 + \right. \right. \\
 & + n^2 \pi^2 h^2 \sin(n\pi)^4 + n^2 \pi^2 h^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^4 + S^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^4 + S^2 \sin(n\pi)^4 - \\
 & - h^2 n^2 \pi^2 \cos(n\pi)^2 \sin(n\pi)^2 \left( e^{\frac{n\pi h}{S}} \right)^4 - 2h^2 n^2 \pi^2 \cos(n\pi)^2 \left( e^{\frac{n\pi h}{S}} \right)^2 + 2hn^2 \pi^2 S \sin(n\pi)^2 - \\
 & - 2hn^2 \pi^2 \left( e^{\frac{n\pi h}{S}} \right)^4 S \sin(n\pi)^2 - h^2 n^2 \pi^2 \cos(n\pi)^2 \sin(n\pi)^2 - 2S^2 \sin(n\pi)^4 \left( e^{\frac{n\pi h}{S}} \right)^2 + \\
 & \left. \left. + h^2 n^4 \pi^4 + 2h^2 n^4 \pi^4 \left( e^{\frac{n\pi h}{S}} \right)^2 + h^2 n^4 \pi^4 \left( e^{\frac{n\pi h}{S}} \right)^4 \right) \right).
 \end{aligned}$$

**Задача 2.** Рассмотрим длинный в направлении  $x$  трехмерный массив, нагруженный постоянной нагрузкой  $Q$  в направлении этой оси, а также оси  $y$ . Размеры массива: длина (ось  $x$ ) –  $S$ , ширина (по оси  $y$ ) –  $R$ , толщина (ось  $z$ ) –  $H$ .

Выбираются начальные функции в виде:

$$\begin{aligned}
 U_0(x, y) &= \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} A_{nm} \sin \frac{n\pi}{S} x \cos \frac{m\pi}{R} y, \\
 V_0(x, y) &= \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} B_{nm} \cos \frac{n\pi}{S} x \sin \frac{m\pi}{R} y, \\
 W_0(x, y) &= \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} C_{nm} \cos \frac{n\pi}{S} x \sin \frac{m\pi}{R} y.
 \end{aligned}$$

На языке системы они записываются в следующем виде:

> Uo:=Sum(Sum(Anm\*sin(a[n]\*x)\*cos(b[m]\*y), m=1.. p), n=1.. p);  
 > Vo:=Sum(Sum(Bnm\*cos(a[n]\*x)\*sin(b[m]\*y), m=1.. p), n=1.. p);  
 > Wo:=Sum(Sum(Cnm\*sin(a[n]\*x)\*sin(b[m]\*y), m=1.. p), n=1.. p);

Результаты для коэффициентов  $A_{nm}$ ,  $B_{nm}$ ,  $C_{nm}$ , выведенные системой в трехмерном случае:

$A_{nm} = 0;$   
 $B_{nm} = 0;$

$C_{nm} = (4QS^2R^2\gamma(-1 + \nu + \cos(m\pi) - \cos(m\pi)\nu + \cos(n\pi)\nu + \cos(n\pi)\nu - \cos(n\pi)\cos(m\pi) + \cos(n\pi)\cos(m\pi)\nu) / (\pi^4 nm(\gamma h \cosh(\gamma h)n^2 R^2 + h \cosh(\gamma h)\gamma m^2 S^2 - \sinh(\gamma h)n^2 R^2 - \sinh(\gamma h)m^2 S^2)).$

Далее в общее численное решение подставляются эти коэффициенты и строятся циклы, в которых производится суммирование. ЭВМ оперирует с дискретным набором данных, в суммах ограничиваем число членов до двадцати. Результаты аналитического и числового решения представлены в работах [6], [8].

Для полупространства решались задачи Фламана и Черутти.

**Задача 3.** Пусть изотропная среда занимает бесконечную область, ограниченную прямой. Такое полубесконечное тело называется полуплоскостью с границей. Рассмотрим задачу о действии нагрузки на границу полуплоскости  $y \geq 0$ :

$$\tau_{xy}|_{y=0} = P(x, \xi),$$

$$\sigma_y|_{y=0} = R(x, \xi).$$

При заданных условиях компоненты напряженно-деформированного состояния не зависят от координаты  $w$  по  $z$ , а перемещение  $w \equiv 0$ . Имеем плоское напряженно-деформированное состояние и плоскую деформацию. Краевая задача замыкается условием обращения в ноль компонентов напряженно-деформированного состояния на бесконечности.

Для задачи Черутти  $P(x, \xi) = \delta(x - \xi)P$  – сосредоточенная сила,  $R(x, \xi) = \delta(x - \xi)R$ ,  $R = 0$ .

Результат решения задачи Черутти для напряжений на Maple:

$$\text{sigmay} := \frac{y^2(-2x + 2\xi)}{\pi((x - \xi)^2 + y^2)^2};$$

$$\text{sigmax} := -\frac{2(x^3 - 3x^2\xi + 3x\xi^2 - \xi^3)}{\pi(x^2 - 2x\xi + \xi^2 + y^2)^2};$$

$$\text{tauxy} := \frac{-\frac{2y}{x^2 - 2x\xi + \xi^2 + y^2} + \frac{2y^3}{(x^2 - 2x\xi + \xi^2 + y^2)^2}}{\pi}.$$

В случае задачи Фламана  $R(x, \xi) = \delta(x - \xi)R$ ,  $P(x, \xi) = \delta(x - \xi)P$ ,  $P = 0$ .

Решение на Maple для напряжений:

$$\text{sigmay} := \frac{2y^3}{\pi((x - \xi)^2 + y^2)^2};$$

$$\text{sigmax} := \frac{2y(x^2 - 2x\xi + \xi^2)}{\pi(x^2 - 2x\xi + \xi^2 + y^2)^2};$$

$$\text{tauxy} := \frac{\frac{2y^2x}{(x^2 - 2x\xi + \xi^2 + y^2)^2} - \frac{2y^2\xi}{(x^2 - 2x\xi + \xi^2 + y^2)^2}}{\pi}.$$

## ВЫВОДЫ

Таким образом, в статье была охарактеризована инструментальная система, осуществляющая процесс моделирования решения задач теории упругости для плиты, полуплоскости в двумерной и массива в трехмерной постановке. Результатами работы системы являются решения для прогибов двумерного и трехмерного тел. При программировании аналитических операций возникли сложности технического характера, связанные с недоработкой вычислительных средств Maple. Внедрением новых программных продуктов в ядро системы эти сложности были устранены.

## СПИСОК ЛИТЕРАТУРЫ

1. Аладьев В. З. Системы компьютерной алгебры: Maple: Искусство программирования / Аладьев В. З. – М. : Лаборатория базовых знаний, 2006. – 792 с.
2. Кабулов В. К. Алгоритмическая система для решения задач теории упругости и пластичности, Сб. / И. К. Кабулов, В. А. Толоч // Вопросы вычислительной и прикладной математики. – 1970. – № 4. – С. 37–45.
3. Гоменюк С. И. Объектно-ориентированные модели и методы анализа механических процессов / С. И. Гоменюк. – Никополь : Никопольская коммунальная типография, 2004. – 316 с.
4. Матросов А. В. Mapleб. Решение задач высшей математики и механики / Матросов А. В. – СПб. : БХВ-Петербург, 2002. – 528 с.
5. Овский А. Г. Использование системы Maple при реализации метода начальных функций Власова / Е. Е. Галан, Овский А. Г., В. А. Толоч // Вісник Запорізького національного університету : Збірник наукових статей. Фізико-математичні науки. – 2008. – № 1. – С. 16–26.
6. Овский А. Г. Моделирование схемы решения трехмерной задачи теории упругости в системе Maple / А. Г. Овский, В. О. Толоч // Гідроакустичний журнал. – 2008. – № 3. – С. 88–97.
7. Власов В. З. Балки плиты и оболочки на упругом основании. / В. Власов, Н. Леонтьев – Москва : ФИЗМАТГИЗ, 1960. – 491 с.
8. Толоч В. А. Операторно-символьные ряды Власова В. З. в решении задач теории упругости в системе Maple / В. В. Шапар, В. А. Толоч // Гідроакустичний журнал. – 2006. – № 3. – С. 66–74.
9. Овский А. Г. Использование системы компьютерной математики Maple для доказательства закона ортогональности матриц прямого и обратного преобразования, составленных Власовым В. З. / А. Г. Овский, В. А. Толоч // Радіоелектроніка, інформатика, управління. – 2008. – № 1. – С. 78–85.
10. Горшков А. Г. Теория упругости и пластичности : учеб. для вузов / Горшков А. Г., Старовойтов Э. И., Талаковский Д. В. – М. : ФИЗМАТЛИТ, 2002. – 416 с.

Надійшла 1.07.2009  
Після доробки 12.05.2010

Овський О. Г., Толоч В. А.

ІНСТРУМЕНТАЛЬНА СИСТЕМА РОЗВ'ЯЗАННЯ ДВОВИМІРНИХ І ТРИВИМІРНИХ ЗАДАЧ ТЕОРІЇ ПРУЖНОСТІ НА БАЗІ MAPLE

У статті наведено опис інструментальної системи розв'язання двовимірних і тривимірних задач теорії пружності. Обчислювальне ядро системи складає Maple. Здійснено формалізацію основних алгоритмів з метою знаходження операцій програмування вказаних задач.



Як приклади роботи з системою наведено розв'язки конкретних задач.

**Ключові слова:** препроцесор, процесор, постпроцесор, метод початкових функцій.

Ovsky A., Tolok V.

DEVELOPMENT SYSTEM FOR SOLVING TWO-DIMENSIONAL AND THREE-DIMENSIONAL ELASTICITY THEORY PROBLEMS BASED ON MAPLE

The authors describe the development system for solving two-dimensional and three-dimensional problems of the theory of elasticity. Maple is a computing kernel of the system. The basic algorithms are formalized for the purpose of finding programming operations of the specified problems. Solutions of specific problems are given as examples of system operation.

**Key words:** preprocessor, processor, postprocessor, method of initial functions.

УДК 004.93

Олійник А. О.

*Канд. техн. наук, старший викладач Запорізького національного технічного університету*

## МУЛЬТИАГЕНТНИЙ МЕТОД З ОПОСЕРЕДКОВАНИМ ЗВ'ЯЗКОМ МІЖ АГЕНТАМИ

Досліджено метод оптимізації на основі мультиагентного підходу. Розроблено мультиагентний метод з опосередкованим зв'язком між агентами, що передбачає виконання додаткових етапів відбору, схрещування та мутації. Проведено експерименти по розв'язанню задач оптимізації багатовимірних нелінійних функцій, що дозволило дослідити ефективність запропонованого методу.

**Ключові слова:** агент, генетичні оператори, мультиагентний метод, оптимізація, цільова функція.

### ВСТУП

Останнім часом широкого застосування набули напрямки штучного інтелекту, основні ідеї яких узяті з природи. До таких напрямків можна віднести: штучні нейронні мережі [1, 2], методи еволюційного пошуку [3, 4], методи оптимізації, засновані на моделюванні колективного інтелекту соціальних тварин [5, 6], штучні імунні системи [7] та ін. [8–13]. Суттєвими перевагами таких напрямків штучного інтелекту є можливість їхнього застосування до розв'язання слабоформалізованих задач та здатність до видобування знань з даних.

Зокрема, мультиагентні методи інтелектуальної оптимізації не висувають додаткових вимог до виду цільової функції, що дозволяє їх застосовувати при розв'язанні оптимізаційних задач, які виникають при побудові моделей складних багатовимірних об'єктів, процесів та систем (на етапах відбору інформативних ознак, структурно-параметричного синтезу моделей, спрощення побудованих моделей). Крім того, на відміну від градієнтних методів багатовимірної нелінійної оптимізації, що традиційно застосовуються при побудові моделей, мультиагентні методи є методами глобальної оптимізації та не схильні до заціклення у локальних оптимумах.

© Олійник А. О., 2010

Для розв'язання завдань дискретної та безперервної оптимізації ефективно застосовується метод оптимізації на основі моделювання переміщення бактерій (Bacteria Foraging Optimization, BFO) [11]. Однак моделі поведінки живих істот, що лежать в основі роботи даного методу, є неповними та не зовсім відповідають своїм природним аналогам. Зокрема, в методі BFO не досить ефективним є механізм утворення нової множини рішень, що призводить до тривалих розрахунків при виконанні оптимізаційного процесу [11].

Тому актуальною є розробка доповнень існуючих моделей, що використовуються для моделювання поведінки живих істот, які забезпечують більш повну відповідність природним аналогам і кращу ефективність роботи зазначених методів оптимізації.

### ПОСТАНОВА ЗАВДАННЯ

Метод BFO ефективно застосовується для розв'язання завдань оптимізації [11], однак математичні моделі, розроблені для даного методу, навіть на думку їх розробників [11] є неповними та не досить ефективними. Тому необхідно доробити відомі математичні моделі BFO для більш адекватного моделювання еволюційних процесів життєдіяльності бактерій.