

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. Н.Э. БАУМАНА

Курсовая работа по методам оптимизаций
Балка под действием сосредоточенной силы

Мещанинов Н.
ФН2-62.

Руководитель: Зарубин В. С.

Москва 2004

1 Постановка задачи

Балка переменного сечения шарнирно оперта по концам и нагружена в середине пролета поперечно направленной сосредоточенной силой P . Длина балки l , плотность материала ρ , его модуль упругости E . Оптимизировать профиль поперечного сечения балки по ее длине исходя из условий минимума массы балки при ограничении на ее наибольший прогиб и из условия минимума наибольшего прогиба при ограничении массы. Сравнить формы упругой линии балки для указанных оптимальных вариантов.

2 Математическая постановка

Рассмотрим первый критерий минимизации, т.е. из условия минимума массы при ограничениях на наибольший прогиб.

Как известно из курса сопротивления материалов [2], уравнение упругой линии нагруженного поперечной силой стержня длиной l определяется уравнением

$$E \frac{d^2}{dx^2} I(x) \frac{d^2 u(x)}{dx^2} = P, \quad x \in (0, l),$$

где $I(x)$ - переменный по длине момент инерции сечения стержня; и граничными условиями

$$u(0) = u_0, u''(0) = u''_0, u(l) = u_l, u''(l) = u''_l.$$

Исходя из постановки задачи, уравнение и условия на границе можно записать следующим образом

$$E \frac{d^2}{dx^2} I(x) \frac{d^2 u(x)}{dx^2} = P \delta \left(x - \frac{l}{2} \right), \quad x \in (0, l), \quad (1)$$

$$u(0) = 0, u''(0) = 0, u(l) = 0, u''(l) = 0.$$

Если обозначить через $r(x)$ профиль балки, переменный по длине, то массу стержня, если любое его сечение представляет собой окружность, можно записать следующим образом

$$m = \pi \rho \int_0^l r^2(x) dx. \quad (2)$$

Таким образом, задачу можно переформулировать так: найти такую функцию $r(x)$, при которой

$$\begin{cases} m \longrightarrow \min \\ \max_{x \in (0, l)} u(x) < \Psi, \end{cases} \quad (3)$$

где Ψ - ограничение на максимальный прогиб, $u(x)$ определяется уравнением (1), а масса m - (2).

Во втором случае, т.е. в случае оптимизации из условия минимума максимального прогиба при ограничениях на массу стержня, получим следующую вариационную задачу

$$\begin{cases} u(x) \longrightarrow \min \\ \pi\rho \int_0^l r^2(x)dx < \Upsilon, \end{cases} \quad (4)$$

где $u(x)$ по-прежнему определяется выражением (1), Υ - максимально допустимая масса стержня.

3 Решение

3.1 Сведение к задаче нелинейного программирования

Поскольку ограничение, вводимое в задачу (3), достаточно неудобно при решении методами вариационного исчисления, то желательно превратить ее в задачу нелинейного программирования и решить численно. Для этого необходимо снизить размерность оптимизации с бесконечности до конечного числа. Это можно сделать, заменив профиль балки $r(x)$ интерполянтном $r_n(x)$, зависящим от n параметров, которые и определяют размерность оптимизации. Так, например, возможен следующий вариант:

$$r_n(x) = a_0 + \sum_{i=1}^{n-1} a_i \sin ix,$$

т.е. заменить исходный профиль балки частичной суммой Фурье и подбирать коэффициенты a_i в соответствии с задачей (3). Такой подход является в некотором смысле универсальным, поскольку позволяет искать решение в классе кусочно-гладких функций. Недостатком же его является требование слишком большой размерности для сколь-либо приемлемой интерполяции исходного профиля. Поэтому ограничим область поиска только гладкими функциями, тогда $r(x)$ удобно заменить следующим интерполянтном

$$r_n(x) = \sum_{i=0}^{n-1} a_i x^i, \quad (5)$$

т.е. частичной суммой степенного ряда.

В таком случае, если \vec{a} - следующий вектор

$$\vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-1} \end{pmatrix},$$

т.е. вектор параметров оптимизации, то введем следующие векторные функции

$$m(\vec{a}) = \pi \rho \int_0^l \left(\sum_{i=0}^{n-1} a_i x^i \right)^2 dx \quad (6)$$

и

$$I(\vec{a}, x), \quad u(\vec{a}, x),$$

где $I(\vec{a}, x)$ - момент инерции сечения стержня, зависящий от его профиля и, естественно, от координаты, а $u(\vec{a}, x)$ получается из аналогичного (1) уравнения

$$E \frac{d^2}{dx^2} I(\vec{a}, x) \frac{d^2 u(\vec{a}, x)}{dx^2} = P \delta \left(x - \frac{l}{2} \right), \quad x \in (0, l), \quad (7)$$

с прежними граничными условиями.

Из всего вышеизложенного следует следующая задача нелинейного программирования: найти такую точку $\vec{a} \in \mathbb{R}^n$, что выполнены следующие условия

$$\begin{cases} m(\vec{a}) \longrightarrow \min \\ \max_{x \in (0, l)} u(\vec{a}, x) < \Psi. \end{cases} \quad (8)$$

3.2 Выбор метода оптимизации

Вид ограничения в задаче (8) таков, что, пожалуй, наиболее рациональным будет использование метода штрафных функций, поскольку большинство других методов (например, метод проекции на множество или проекции антиградиента) требуют явного указания допустимого множества Ω , и, кроме того, это множество должно быть выпуклым. В нашем случае, т.е. в случае задачи (8), допустимым множеством является множество тех точек $\vec{a} \in \mathbb{R}^n$, при которых выполняется неравенство

$$\max_{x \in (0, l)} u(\vec{a}, x) < \Psi. \quad (9)$$

Доказательство выпуклости такого множества само по себе является нетривиальной задачей, а явная запись Ω в общем случае вообще не представляется возможным.

3.3 Метод штрафных функций

Метод же штрафных функций заключается в безусловной минимизации

$$f(x) + \delta_{\Omega}(x) \longrightarrow \min,$$

где $f(x)$ - целевая функция, а $\delta_\Omega(x)$ - индикаторная функция

$$\delta_\Omega(x) = \begin{cases} 0, & x \in \Omega; \\ +\infty, & x \notin \Omega. \end{cases}$$

Каждое из этих неравенств (а это именно неравенства, т.к. ограничение имеет вид (9)) можно легко проверить.

Вместо функции $\delta_\Omega(x)$ применяют последовательность функций $\delta_k(x)$, таких, что

$$\lim_{k \rightarrow \infty} \delta_k(x) = \delta_\Omega(x). \quad (10)$$

Это нужно для того, чтобы "сгладить" разрыв функции $\delta_\Omega(x)$, который не позволяет запускать методы безусловной минимизации. Функции $\delta_k(x)$ могут быть выбраны по-разному [1], в частности, следующим образом

$$\delta_k(x) = \begin{cases} 0, & x \in \Omega; \\ > 0, & x \notin \Omega. \end{cases}$$

Функция такого вида называется функцией внешнего штрафа. Если множество Ω задано в виде неравенства, то обычно используют в качестве функции штрафа следующую функцию [1]

$$\delta_k(x) = r_k \left(\frac{|g(x)| + g(x)}{2} \right)^2,$$

где $g(x)$ определяет ограничение $g(x) \leq 0$, а для того, чтобы выполнялось условие (10), необходимо следующее условие: $r_k \rightarrow \infty$. Окончательно, "объектом" минимизации на каждой итерации с номером k безусловного метода выступает функция

$$f_k(x) = f(x) + \delta_k(x) = f(x) + r_k \left(\frac{|g(x)| + g(x)}{2} \right)^2.$$

Если в задаче m ограничений, то это выражение легко обобщается

$$f_k(x) = f(x) + \delta_k(x) = f(x) + r_k \sum_{i=1}^m \left(\frac{|g_i(x)| + g_i(x)}{2} \right)^2.$$

3.4 Реализация алгоритма метода

Используя обозначения, введенные в предыдущем разделе, можно записать выражение для штрафной функции применительно к исходной задаче. Для начала заметим, что решение нашей задачи требует на каждом шаге метода безусловной оптимизации численного решения дифференциального уравнения (7). Это означает, что вместо функции $u(\vec{a}, x)$

мы будем иметь p -мерный вектор $\vec{u}(\vec{a})$

$$\vec{u}(\vec{a}, x) = \begin{pmatrix} u_0(\vec{a}) \\ u_1(\vec{a}) \\ \dots \\ u_{p-1}(\vec{a}) \end{pmatrix} = \begin{pmatrix} 0 \\ u_1(\vec{a}) \\ \dots \\ u_{p-2}(\vec{a}) \\ 0 \end{pmatrix}.$$

Поэтому естественно ввести функцию $M(\vec{u})$, находящую максимальный элемент этого вектора

$$M(\vec{u}) = \max_{i=0, p-1} u_i(\vec{a}).$$

В таком случае, выражение для $\delta_k(x)$ можно записать следующим образом

$$\delta_k(x) = r_k \left(\frac{|M(\vec{u}) - \Psi| + M(\vec{u}) - \Psi}{2} \right)^2,$$

а выражение для всей штрафной функции так

$$f_k(\vec{a}) = m(\vec{a}) + r_k \left(\frac{|M(\vec{u}) - \Psi| + M(\vec{u}) - \Psi}{2} \right)^2,$$

или, учитывая выражение (6),

$$f_k(\vec{a}) = \pi\rho \int_0^l \left(\sum_{i=0}^{n-1} a_i x^i \right)^2 dx + r_k \left(\frac{|M(\vec{u}) - \Psi| + M(\vec{u}) - \Psi}{2} \right)^2. \quad (11)$$

В качестве метода безусловной минимизации был выбран алгоритм Хука-Дживса, использующего в качестве вспомогательной задачи исчерпывающего спуска метод дихотомии. Параметры r_k выбирались в зависимости от размерности оптимизации, поскольку одно и то же множество $\{r_k\}_{k=0}^{\infty}$ не обеспечивало надежного "барьера" при различных значениях n .

ПРИМЕЧАНИЕ. Выпишем выражение для $f_k^*(\vec{a})$ для задачи (4), которая аналогично (3) сводится к задаче нелинейного программирования: найти такую точку $\vec{a} \in \mathbb{R}^n$, что выполнены следующие условия

$$\begin{cases} \max_{x \in (0, l)} u(\vec{a}, x) \longrightarrow \min \\ m(\vec{a}) < \Upsilon \end{cases}. \quad (12)$$

С одной стороны, $f_k^*(\vec{a})$ могла быть записана следующим образом

$$f_k^*(\vec{a}) = M(\vec{u}) + r_k \left(\frac{\left| \pi\rho \int_0^l \left(\sum_{i=0}^{n-1} a_i x^i \right)^2 dx - \Upsilon \right| + \pi\rho \int_0^l \left(\sum_{i=0}^{n-1} a_i x^i \right)^2 dx - \Upsilon}{2} \right)^2,$$

но, к сожалению, такого вида штрафная функция не позволила быстро подобрать параметры r_k для надежного "заточения" функции f_k^* в допустимое множество Ω . Поэтому, использовалась функция ζ_k с более "крутыми" краями, которая строится следующим образом:

$$\xi(\vec{a}) = \pi\rho \int_0^l \left(\sum_{i=0}^{n-1} a_i x^i \right)^2 dx - \Upsilon,$$

$$\xi(\vec{a}) = \frac{|\xi(\vec{a})| + \xi(\vec{a})}{2},$$

$$\zeta_k(\vec{a}) = M(\vec{u}) + r_k \xi(\vec{a}) e^{\xi(\vec{a})}.$$

3.4.1 Пошаговая инструкция к алгоритму

1. Сгенерировать начальный профиль для старта алгоритма. (Применялся прямой профиль, дающий прямую цилиндрическую балку). Для этого необходимо задать вектор $\vec{a}^{(0)}$. (В случае прямого цилиндрического профиля $a_0^0 \neq 0, a_i^0 = 0, i = \overline{1, n-1}$);
2. Запуск минимизации функции $f_k(\vec{a}^{(k)})$ либо $\zeta_k(\vec{a}^{(k)})$. (Применялся описанный выше алгоритм Хука-Дживса. В качестве критерия останова было выбрано неравенство $\|\vec{a}^{(k)} - \vec{a}^{(k-1)}\|_2 < \varepsilon, \vec{a}_l^{(k)}, \vec{a}_{l-1}^{(k)} \in \mathbb{R}^n$);
 - Перед каждым вычислением функции $f_k(\vec{a}^{(k)})$ ($\zeta_k(\vec{a}^{(k)})$) необходимо обновлять решение $\vec{u}(\vec{a}^{(k)})$ уравнения (7);
 - Если в алгоритме предусмотрен исчерпывающий спуск (в нашем случае это так), то в процедуре, реализующей его, также необходимо обновлять вектор $\vec{u}(\vec{a}^{(k)})$

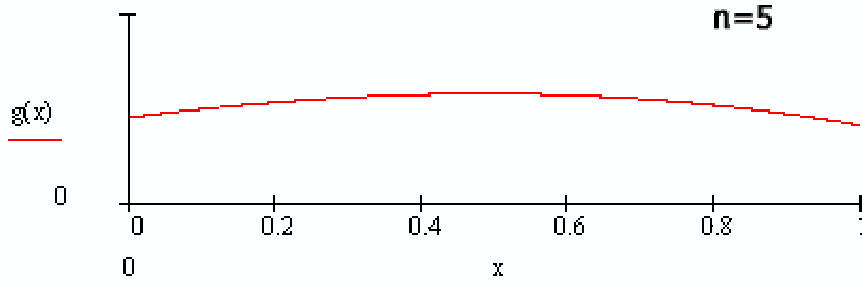
3.5 Результаты

3.5.1 Задача (8)

Решение проводилось при $n = \{3, 4, 5\}$. Результирующий профиль выглядит так, как показано на рисунке 3.5.1. Выпишем результаты в виде таблицы, в которой ε определяет отношение массы, полученной при данной размерности n , к массе при размерности $n - 1$. В случае $n = 3$ оно представляет собой отношение к массе, полученной при одномерной оптимизации кусочно-гладкого профиля с изломом в середине балки.

n	ε	r_0	r_k
3	0,786023	2	$r_k = r_{k-1} + 2$
4	1,009053	1	$r_k = r_{k-1} + 1$
5	1,001279	4	$r_k = r_{k-1} + 4$

Рис. 1: Результирующий профиль при первом типе оптимизации



Во всех случаях дифференциальное уравнение (7) решалось на 1000-точечной сетке при прочих равных параметрах алгоритма Хука-Дживса. Видно, что разница значений двух последних ε отличаются почти на порядок. Увеличение n приводило к увеличению времени счета и не давало принципиально новых результатов. Выпишем значения коэффициентов a_i для различных размерностей оптимизации

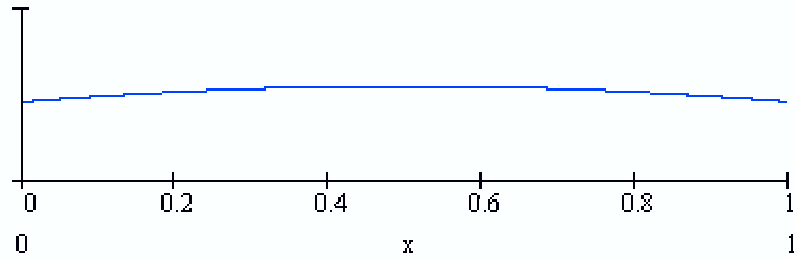
Размерность	Коэффициенты	Количество итераций	время счета
$n = 3$	$a_0 = 0.088569$ $a_1 = 0.122953$ $a_2 = -0.131914$	65	1m 11s
$n = 4$	$a_0 = 0.089906$ $a_1 = 0.135823$ $a_2 = -0.182736$ $a_3 = 0.038686$	259	5m 47s
$n = 5$	$a_0 = 0.091342$ $a_1 = 0.101192$ $a_2 = -0.098367$ $a_3 = -1.094385 \cdot 10^{-3}$ $a_4 = -0.010752$	31	0m 56s

3.5.2 Задача (12)

Решение также проводилось при $n = \{3, 4, 5\}$. Результирующий профиль выглядит так, как показано на рисунке 3.5.2.

Удивительной особенностью решения данной задачи явилось то, что увеличение n не дало вообще никаких изменений, т.е. интерполяция квадратичной функцией дала почти точный результат, т.к. при увеличении размерности оптимизации с $n = 3$ до $n = 4$ коэффициент a_3 в ходе оптимизации обнулялся, а коэффициенты a_i , $i = 1, 2$ совпадали с соответствующими коэффициентами в случае трехмерной минимизации. Аналогичная картина наблюдалась и при $n = 4$.

Рис. 2: Результирующий профиль при втором типе оптимизации



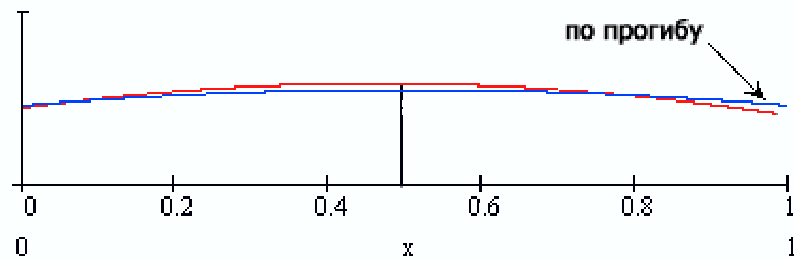
n	ε	r_0	r_k
3	0,821334	2	$r_k = r_{k-1} + 4$
4	1	2	$r_k = r_{k-1} + 4$
5	1	2	$r_k = r_{k-1} + 4$

Таблица коэффициентов выглядит следующим образом

Размерность	Коэффициенты	Количество итераций	время счета
$n = 3$	$a_0 = 0.091567$ $a_1 = 0.073731$ $a_2 = -0.073731$	3	0m 5s
$n = 4$	$a_0 = 0.091567$ $a_1 = 0.073731$ $a_2 = -0.073731$ $a_3 = 0.000000$	3	0m 7s
$n = 5$	$a_0 = 0.091567$ $a_1 = 0.073731$ $a_2 = -0.073731$ $a_3 = 0.000000$ $a_4 = 0.000000$	3	0m 9s

3.5.3 Сравнение двух профилей

Поскольку профили получились весьма похожими, наложим один на другой, чтобы разница была более очевидной. Видно, что при опти-



мизации при условии минимума прогиба, профиль более пологий, что объясняется большим произволом в выборе массы. Поэтому вещество стержня концентрируется вокруг точки приложения силы, что и обеспечивает минимальность наибольшего прогиба. Кроме того, если сравнить максимальный прогиб в случае первого типа оптимизации и второго, то можно записать отношение

$$\eta = \frac{u_{max1}}{u_{max2}} = 0,82984, \quad (13)$$

где u_{max1} - максимальный прогиб при решении задачи (12), а u_{max2} - задачи (8).

Список литературы

- [1] Аттетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизаций. М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 440 с.
- [2] Феодосьев В.И. Сопротивление материалов. М.: Изд-во МГТУ им. Н.Э. Баумана, 1999. 592 с.