Определение деформации балки, закреплённой на обоих концах и нагруженной распределённой нагрузкой

Дана балка, жёстко закреплённая на обоих концах. Деформацию балки под действием заданной системы сил можно найти из следующей системы уравнений:

$$EJ\,\frac{\mathrm{d}^4 u}{\mathrm{d}x^4} \;=\; f(x)\;;$$

$$u(0) \;=\; u(L) \;=\; 0\,, \quad u'(0) \;=\; u'(L) \;=\; 0\;. \tag{1}$$

Здесь EJ – изгибная жёсткость балки, $u(x)$ – прогиб её в точке с координатой x, $f(x)$ – функция, задающая действующую на балку распределённую нагрузку.

Математической задаче (1) можно придать другую формулировку, что мы сейчас и сделаем.

Пусть V – пространство функций, заданных на области $\Omega = [\,0\,,L\,]$, имеющих кусочно-непрерывные 1-ю и 2-ю производные и удовлетворяющих нулевым граничным условиям:

$$V \;=\; \{\; v:\; v \in C^{(2)}(\Omega),\;\; v(0) = v(L) = 0,\;\; v'(0) = v'(L) = 0 \;\}\;.$$

Умножим первое уравнение (1) на v и дважды проинтегрируем по частям, в результате чего получим:

$$\int_0^L EJu''v''\,\mathrm{d}x \;=\; \int_0^L f\,v\,\mathrm{d}x \qquad \forall v \in V. \tag{2}$$

Далее, обозначим:

$$a(u,v) \;=\; \int_0^L EJu''v''\,\mathrm{d}x\,, \quad b(f,v) \;=\; \int_0^L f\,v\,\mathrm{d}x\,,$$

$$J(v) \;=\; a(u,v) - b(f,v)\;.$$

Тогда получим:

$$J(u) \;\leqslant\; J(v)\,, \qquad \forall v \in V. \tag{3}$$

Все три формулировки $(1)-(3)$ эквивалентны.

Отметим, что в формулировке (3) $J(u)$ интерпретируется как полная энергия (т.е. внутренняя плюс потенциальная энергия внешних сил), а формулировка (2) – прямое выражение принципа возможных перемещений.

Разобьем отрезок $[0, L]$ на $n+1$ подотрезков (это и будут наши конечные элементы).

Положим $h = \dfrac{L}{n+1}$. Концы подотрезков служат узлами сетки

$$\Delta = \{X_i\}_{i=0}^{i=n+1}, \qquad X_i = ih, \qquad i = 0, \ldots, n+1.$$

Будем искать приближенное решение задачи (1) в классе эрмитовых кубических многочленов, т.е. в пространстве $\mathbb{P}_{3,2}^{\Delta}$ (речь идёт о кусочных многочленах с узлами из сетки Δ, имеющих степень 3 и дефект 2). Однако нас устраивают лишь те многочлены $v(x)$, которые удовлетворяют граничным условиям:

$$v(0) = v(L) = 0, \quad v'(0) = v'(L) = 0. \tag{4}$$

Такие кусочные многочлены образуют подпространство V_Δ в $\mathbb{P}_{3,2}^{\Delta}$:

$$V_\Delta \subset \mathbb{P}_{3,2}^{\Delta}, \qquad V_\Delta \in V.$$

Нам понадобится выбрать определённый базис в пространстве V_Δ, а сначала мы выясним, какова размерность этого пространства. Учтём, что

$$\dim \mathbb{P}_{3,2}^{\Delta} = m + 1 + r(n-1).$$

Здесь m – степень многочлена, r – дефект, n – число интервалов разбиения. В рассматриваемом нами случае $m = 3$, $r = 2$ и вместо n взят $n+1$ интервал. Итак:

$$\dim V_\Delta = \dim \mathbb{P}_{3,2}^{\Delta} - 4 = 3 + 1 + 2n - 4 = 2n$$

(именно поэтому исходный отрезок и был разбит на $n+1$ подотрезков).

Таким образом, нужно $2n$ базисных функций. Будем их вводить, стараясь сделать носитель (т.е. замыкание множества точек, на котором функция отлична от нуля) каждой из них по возможности меньшим.

В качестве n базисных функций используем такие:

$$W_i(x) = (\, x \leqslant x_L \,?\; 0 : x \leqslant x_M \,?\; W_i^L(x) : x \leqslant x_D \,?\; W_i^D(x) : 0\,),$$

где x_L – левый узел носителя, т.е. $x_L = x_i$;

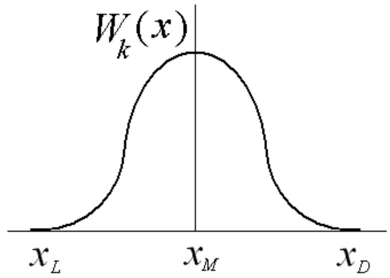
x_M – средний узел носителя, т.е. $x_M = x_i + h$;

x_D – правый узел носителя, т.е. $x_D = x_i + 2h$;

$$W_i^L(x) \; = \; -\frac{(x-x_L)^2}{h^3}\,[\,(x-x_M)+(x-x_D)\,]\,,$$

$$W_i^D(x) \; = \; \frac{(x-x_D)^2}{h^3}\,[\,(x-x_M)+(x-x_L)\,]\,.$$

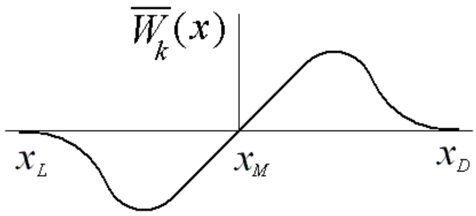
Эти функции внешне напоминают B-сплайны, но носитель каждой из них $-$ в два раза меньше, чем у B-сплайна (содержит 2 интервала разбиения, а не 4).



Теперь возьмём ещё n базисных функций:

$$\overline{W}_i(x) \; = \; (\; x \leqslant x_L \,?\;\; 0 \,:\; x \leqslant x_M \,?\;\; \overline{W}_i^L(x) :\; x \leqslant x_D \;?\;\; \overline{W}_i^D(x) \,:\; 0\;)\,,$$

где $\overline{W}_i^L(x) \; = \; \dfrac{(x-x_L)^2}{h^2}\,(x-x_M)\,, \quad \overline{W}_i^D(x) \; = \; \dfrac{(x-x_D)^2}{h^2}\,(x-x_M)\,.$



Итак, любую функцию $\upsilon(x) \in V_\Delta$ можно представить в виде

$$\upsilon(x) \; = \; \upsilon_0\, W_0(x) + \overline{\upsilon}_0\, \overline{W}_0(x) + \ldots + \overline{\upsilon}_{n-1}\, \overline{W}_{n-1}(x)\,.$$

В рассматриваемом случае решение будем искать в виде

$$u(x) \; = \; \sum_{i=0}^{i<n} (\, u_i\, W_i(x) + \overline{u}_i\, \overline{W}_i(x)\,)\,; \qquad (5)$$

коэффициенты $u_i, \overline{u}_i$ нам и надо определить.

Замечание. Многочлены $W_i^L(x)$, $W_i^D(x)$, $\overline{W}_i^L(x)$, $\overline{W}_i^D(x)$, из которых строятся базисные функции, получаются из функций Кунса ($K_1(s)$, $K_0(s)$, $K_3(s)$, $K_2(s)$), если преобразовать последние к отрезкам длины h и ввести нормировку:

$$W_i^L(x_{i+1}) = W_i^D(x_{i+1}) = 1 \text{ и } \overline{W}_i^{L\,\prime}(x_{i+1}) = \overline{W}_i^{D\,\prime}(x_{i+1}) = 1 .$$

Вместо условия (2) потребуем, чтобы

$$\int_0^L EJ\,u''v''\,\mathrm{d}x = \int_0^L f\,v\,\mathrm{d}x \qquad \forall\, v \in V_\Delta , \tag{2°}$$

т.е. $a(u,v) = b(f,v) \quad \forall\, v \in V_\Delta$.

Для этого необходимо и достаточно, чтобы

$$a(u, W_i) = b(f, W_i) , \quad a(u, \overline{W}_i) = b(f, \overline{W}_i) , \tag{6}$$
$$i = 0 , \ldots , n-1 .$$

Итак, имеем $2n$ уравнений – столько, сколько коэффициентов в выражении (5). Подставляя (5) в (6), получим систему линейных алгебраических уравнений (СЛАУ):

$$\sum_{i=0}^{i<n} [\, a(W_i, W_k)\, u_i + a(\overline{W}_i, W_k)\, \bar{u}_i \,] = b(f, W_k) ,$$

$$\sum_{i=0}^{i<n} [\, a(W_i, \overline{W}_k)\, u_i + a(\overline{W}_i, \overline{W}_k)\, \bar{u}_i \,] = b(f, \overline{W}_k) .$$

Как вычислять коэффициенты этой СЛАУ? Представим $(2°)$ в виде суммы

$$\sum_{i=0}^{i<n} [\int_{x_j}^{x_{j+1}} EJ\,u''v''\mathrm{d}x\,] = \sum_{j=0}^{j<n} [\int_{x_j}^{x_{j+1}} f\,v\,\mathrm{d}x\,] .$$

Иными словами, уравнение $(2°)$ есть сумма $n+1$ равенства

$$a^{(j)}(u,v) = b^{(j)}(f,v) .$$

Таким образом, решение поставленной задачи сводится к решению системы линейных алгебраических уравнений (СЛАУ). Эту систему можно представить в стандартном виде $A\,U = B$, причём получается она почленным суммированием $n+1$ элементарных СЛАУ

$$A^{(j)}\,U = B^{(j)} ,$$

записанных для каждого конечного элемента в отдельности.

Элементарные матрицы и столбцы устроены просто (тут сказывается выбор базисных функций с компактным носителем малой длины). С интервалом (x_i, x_{i+1}) непустое пересечение имеют только носители четырех ба зисных функций $W_{i-1}(x)$, $\overline{W}_{i-1}(x)$, $W_i(x)$, $\overline{W}_i\,(x)$ (для крайне левого и правого интервалов будет только две функции).

В матрице $A^{(j)}$ отличен от нуля лишь следующий блок 4×4:

$$\begin{pmatrix} a^{(j)}(W_{j-1}^{D}, W_{j-1}^{D}) & \dots & a^{(j)}(W_{j-1}^{D}, \overline{W}_j^{L}) \\ \dots & \dots & \dots \\ a^{(j)}(\overline{W}_j^{L}, W_{j-1}^{D}) & \dots & a^{(j)}(\overline{W}_j^{L}, \overline{W}_j^{L}) \end{pmatrix}.$$

а в столбце $B^{(j)}$ не равны нулю лишь четыре элемента: $b^{(j)}(f, W_{j-1}^{D})$, $b^{(j)}(f, \overline{W}_{j-1}^{D})$, $b^{(j)}(f, W_j^{L})$, $b^{(j)}(f, \overline{W}_j^{L})$ (здесь вместо самих базисных функций сразу подставлены те многочлены, к которым эти функции сводятся на интервале (x_i, x_{i+1})).

Коэффициенты матрицы $A^{(j)}$ можно вычислить аналитически. Например:

$$a^{(j)}(W_{j-1}^{D}, W_{j-1}^{D}) = \int_{x_j}^{x_{j+1}} EJ\, W_{j-1}^{D\,''}(x)\, W_{j-1}^{D\,''}(x)\,\mathrm{d}x = 12\,\frac{EJ}{h^3}.$$

Аналогично вычисляются и другие её коэффициенты. В итоге получаем:

$$A^{(j)} = \frac{EJ}{h^3}\begin{pmatrix} 12 & 6h & -12 & 6h \\ 6h & 4h^2 & -6h & 2h^2 \\ -12 & -6h & 12 & -6h \\ 6h & 2h^2 & -6h & 4h^2 \end{pmatrix}.$$

Коэффициенты столбца $B^{(j)}$ в общем случае находятся численным интегрированием:

$$b^{(j)}(f, W_{j-1}^{D}) = \int_{x_j}^{x_{j+1}} f(x)\, W_{j-1}^{D}(x)\,\mathrm{d}x.$$

Переход от элементарных СЛАУ $A^{(j)}U^{(j)} = B^{(j)}$, где $U^{(j)}$ – столбец из элементов u_{j-1}, $\overline{u}_{j-1}$, u_j, $\overline{u}_j$, к полной СЛАУ $AU = B$, определяющей решение глобальной задачи, носит название ансамблирования.

Таким образом, при программировании нужно обнулить массивы A и B, а затем в цикле при $j = 0, \dots, n$ добавлять (используя операцию **+=** языка Си) к их элементам значения элементов матрицы $A^{(j)}$ и столбца $B^{(j)}$. При этом полученная в итоге матрица A оказывается симметричной положи тельно определённой ленточной матрицей с полушириной ленты $m = 4$ (полуширина ленточной матрицы – это максимальное число ненулевых элементов её строки, стоящих на диагонали и справа от неё).

Такую СЛАУ можно эффективно решить, используя вариант метода Холецкого, ориентированный на работу с ленточными матрицами. Полученное приближённое решение (5) будет удовлетворять соотношениям

$$u(x_{i+1}) = u_i , \quad u'(x_{i+1}) = \bar{u}_i .$$

Замечание 1. Мы рассмотрели порядок решения краевой задачи (1) с нулевыми граничными условиями. Случай граничных условий общего вида не представляет трудностей, поскольку решение можно искать в виде

$$u(x) = u_1(x) + u_0(x) ,$$

где $u_0(x)$ – функция, удовлетворяющая поставленным граничным условиям, а функция $u_1(x)$ находится изложенным выше образом.

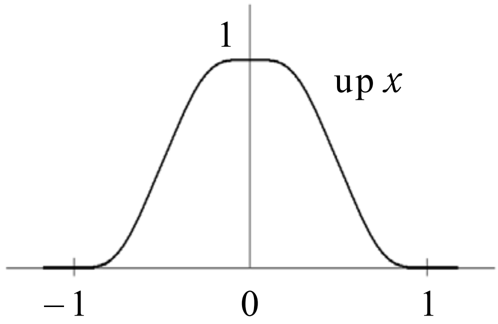
Замечание 2. Приближённое решение исходной задачи было построено в виде эрмитова кубического многочлена. Значит, это – функция класса C^1; её четвёртая производная не определена в узлах сетки Δ, а в остальных точках отрезка $[0, L]$ тождественно равна нулю, так что вовсе не очевидно, что она может служить разумным приближением задачи (1). Однако если взять значения точного решения и его первой производной в узлах x_i и затем по этим значениям на отрезке $[0, L]$ непрерывный интерполянт, то – согласно (3) – данный интерполянт будет относительно точного решения задачи давать бóльшую погрешность, чем построенное нами приближённое решение.

Несколько конкретизируем теперь поставленную задачу. Будем предполагать, что система сил, действующих на балку, сводится к нескольким сосредоточенным силам, приложенным в определённых точках X_k^*. Математически сосредоточенную силу обычно представляют δ-функцией, но мы поступим более реалистично: предположим, что её носитель хотя и мал, но конечен (как это и имеет место в действительности).

Говоря более конкретно, воспользуемся функцией $\mathrm{up}\, x$, которая определяется следующим образом:

$$\mathrm{up}\, x = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-itx} \prod_{k=1}^{\infty} \frac{\sin 2^{-k} t}{2^{-k} t} \, \mathrm{d}t .$$

Носителем этой бесконечно дифференцируемой функции является отрезок $[-1, 1]$, а график её таков:



Если же взять функции $K \operatorname{up} Kx$, то они при $K \to \infty$ сходятся к δ-функции (речь идёт о сходимости в смысле теории обобщённых функций), так что любая из них при достаточно большом K даёт разумную аппроксимацию δ-функции. Носители таких функций при $K \to \infty$ стягиваются к точке 0. Таким образом, для случая двух сосредоточенных сил будем иметь для функции $f(x)$ выражение вида

$$f(x) = A + A_1 K \operatorname{up}(K(x - X_1^*)) + A_2 K \operatorname{up}(K(x - X_2^*)),$$

где A_k − модуль k-й силы, а слагаемое A определяет интенсивность равномерно распределённой нагрузки. В условии конкретного варианта задания данное выражение может (при определённом выборе единиц измерения и конкретных значениях параметров) быть записано, например, в таком виде:

```
-200*(1-300*up(30*(x-0.1))+100*up(30*(x-0.6)))
```

Для решения рассматриваемой задачи может оказаться полезной представляемая здесь типовая программа **Beam_31.c**. Она позволяет последовательно решить данную задачу соответственно для 3, 6, 12 и 24 конечных элементов. Программа написана на языке Си и рассчитана на использование компилятора **Turbo C**. Для внесения модификаций в программу удобно пользоваться интегрированной средой компилятора, а для создания выполнимого файла (**exe**-файла) − пакетной версией этого же компилятора, вызываемой через файл **pgm.bat**.

Модификация типовой программы при переходе к выполнению нового варианта задания ограничивается исправлением номера варианта (переменная **var**), значений L и EJ (переменные **L** и **EJ**) в начальной части программной единицы **main** и выражения, задающего вид функции $f(x)$ в программной единице **Fun** (это же выражение следует продублировать в операторе **printf("Нагрузка: ...\n");**, стоящем в начальной части программной единицы **Problem**).

Помимо программной единицы **main**, программа **Beam_31.c** содержит следующие программные единицы:

- функцию **Fun**, которая, как уже отмечалось, отвечает за вычисление функции $f(x)$;
- процедуру **Problem**, позволяющую найти решение $u(x)$ поставленной задачи для одного конкретного значения n (она четыре раза вызывается из **main**);
- функции **Bas_w** и **Bas_ww**, которые вычисляют значения базисных функций $W_i(x)$ и $\overline{W_i}(x)$;
- функции **Bas_wL**, **Bas_wD**, **Bas_wwL**, **Bas_wwD**, которые вычисляют значения многочленов $W_i^L(x)$, $W_i^D(x)$, $\overline{W}_i^L(x)$, $\overline{W}_i^D(x)$;
- функцию **Solution**, которая вычисляет значение решения $u(x)$ по формуле (3);
- функцию **Alloc**, обращение к которой позволяет выделить память для размещения одномерного массива чисел типа **double** заданной длины;
- процедуру **Sorry**, отвечающую за вывод информационного сообщения в случае нехватки оперативной памяти;
- процедуру **Wait**, при обращении к которой программа приостанавливает работу до нажатия пользователем какой-либо клавиши (ввод комбинации **Ctrl+C** позволяет ему аварийно прервать работу программы);
- процедуры **CholDekB** и **CholSolB**, позволяющие решить методом Холецкого СЛАУ с ленточной матрицей;
- функцию **up**, которая вычисляет значение функции $\mathrm{up}\, x$;
- функцию **UFint**, которая реализует алгоритм адаптивной квадратурной процедуры **QUANC8** и позволяет вычислить значение определённого интеграла от функции одного переменного по заданному отрезку (заметим, что реализация функции **UFint** использует принцип инверсного управления предполагается, что вызов её происходит в цикле, при каждом прохождении которого пользователь должен один раз вычислить значение подынтегральной функции, причём текущее значение аргумента и признак завершения цикла определяются функцией **UFint**);

– процедуры **k_UFint** и **f_UFint**, которые служат соответственно для инициализации функции **UFint** и для завершения работы с ней.

Созданный в результате компиляции программы **Beam_31.c** выполнимый файл **Beam_31.exe** последовательно выдаёт на экран видеомонитора следующие результаты:

– номер варианта и исходные данные для задачи об изгибе балки;
– для каждого n: половину ленты симметричной матрицы A и столбцы B и U (столбцы при этом выводятся в две колонки, представляющие соответственно верхнюю и нижнюю половины этих столбцов);
– таблицу результатов, содержащую значения найденного решения $u(x)$ в 17 точках x_k, равномерно расположенных на отрезке $[0, L]$ (таблица включает шесть колонок, в которые выводятся номер k, значение x_k и четыре полученных значения $u(x_k)$ – соответственно для 3, 6, 12 и 24 конечных элементов).

Следует иметь в виду, что значения $u(x_k)$ (равно как и элементы каждого из столбцов U) выводятся с переходом к более мелким единицам измерения (а именно, вычисленные программой значения при выводе умножаются на 1000).

Всякий раз, как только экран будет заполнен очередной порцией результатов, программа **Beam_31.exe** приостанавливает свою работу, ожидая, пока пользователь для продолжения ей работы не нажмёт какую-либо клавишу (удобно пользоваться клавишей **Esc**).

Может оказаться удобным, чтобы вывод результатов происходил не на экран, а в текстовый файл. Этого легко добиться, применяя стандартную для **DOS** технику переназначения вывода: достаточно набрать в командной строке вместо **Beam_31.exe** следующий текст:

```
Beam_31.exe > result.txt
```

нажав после этого несколько раз **Esc**, вы получите файл **result.txt** с результатами работы программы.