

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДА ХАБОЛТА ДЛЯ ПРЯМОГО ИНТЕГРИРОВАНИЯ

Борщёв А.И.,

Научный руководитель – Напрасников В.В., к.т.н., доцент

Целью данной работы реализация программная реализация метода Хаболта для прямого интегрирования уравнений равновесия в динамических задачах. В качестве среды для программной реализации методов выбран Mathcad 15.

Уравнение равновесия системы конечных элементов, находящейся в состоянии движения, имеет вид:

$$F_I(t) + F_D(t) + F_E(t) = R(t),$$

где $F_I(t)$ — вектор сил инерции, $F_I(t) = M\ddot{U}$; $F_D(t)$ вектор сил демпфирования, $F_D(t) = C\dot{U}$; $F_E(t)$ — вектор сил упругости, $F_E(t) = KU$, M , C и K соответственно матрицы масс, демпфирования и жесткости; R — вектор внешней узловой нагрузки, U , $\dot{U}$ и $\ddot{U}$ — векторы узловых перемещений, скоростей и ускорений ансамбля конечных элементов, все векторы зависят от времени.

Таким образом, в динамических задачах рассматривается статическое равновесие системы в момент времени t с учетом инерционных сил, зависящих от ускорения, и сил демпфирования, зависящих от скоростей движения. На практике такие задачи решаются с помощью методов прямого интегрирования.

Прямое численное интегрирование основано на двух идеях.

Во-первых, удовлетворение условий равновесия требуется не в любой момент времени t , а только на отдельных коротких отрезках времени Δt . Это означает, что равновесие с учетом сил инерции и демпфирования рассматривается в дискретных точках временного интервала.

Во-вторых, учитывается изменение перемещений, скоростей и ускорений внутри каждого временного интервала Δt .

Пошаговое интегрирование методом Хаболта:

А. Начальные вычисления:

1. Формируются матрицы жесткости K , масс M , демпфирования C
2. Задаются начальные значения U_0 , $\dot{U}_0$ и $\ddot{U}_0$
3. Выбирается временной шаг Δt и вычисляются постоянные интегрирования:

$$a_0 = \frac{2}{\Delta t^2}; \quad a_1 = \frac{11}{6\Delta t}; \quad a_2 = \frac{25}{\Delta t^2}; \quad a_3 = \frac{3}{\Delta t}; \quad a_4 = -2a_0;$$

$$a_5 = \frac{-a_3}{2}; a_6 = \frac{a_0}{2}; a_7 = \frac{a_3}{9}$$

4. Вычисляются $\dot{U}_{\Delta t}$ и $U_{2\Delta t}$ с помощью специальной начальной процедуры

5. Вычисляется эффективная матрица жесткости

$$\hat{K} = K + a_0 M + a_1 C$$

6. Матрица $\hat{K}$ приводится к треугольному виду

Б. Для каждого шага

1. Вычисляется вектор эффективной нагрузки в момент

$$\hat{R}_{t+\Delta t} = R_{t+\Delta t} + M(a_2 U_t + a_4 U_{t-\Delta t} + a_6 U_{t-2\Delta t}) + C(a_3 U_t + a_5 U_{t-\Delta t} + a_7 U_{t-2\Delta t})$$

2. Определяются переменные в момент времени

$$LDL^T U_{t+\Delta t} = \hat{R}_{t+\Delta t}$$

3. Если требуется, вычисляются скорости и ускорения для момента

$$\ddot{U}_{t+\Delta t} = a_0 U_{t+\Delta t} - a_2 U_t - a_4 U_{t-\Delta t} - a_6 U_{t-2\Delta t}$$

$$\dot{U}_{t+\Delta t} = a_1 U_{t+\Delta t} - a_3 U_t - a_5 U_{t-\Delta t} - a_7 U_{t-2\Delta t}$$

Программная реализация численного метода:

```

Habolit(K,M,C,dt,R_tdt,U_t,U_tdt,U_2dt) :=
a0 ← 2/dt^2, a1 ← 11/6dt, a2 ← 5/dt^2, a3 ← 3/dt, a4 ← -2a0, a5 ← -a3/2, a6 ← a0/2, a7 ← a3/9
K_hat ← K + a0·M + a1·C  Вычисление эффективной матрицы жесткости
T ← ("t" "U1" "U2")
T ← stack[T, (dt U_2dt_0 U_t2dt_1)]
T ← stack[T, (2·dt U_tdt_0 U_tdt_1)]
T ← stack[T, (3·dt U_t_0 U_t_1)]
for i ∈ 4..N
  R_eff ← R_tdt + M·(a2·U_t + a4·U_tdt + a6·U_2dt) + C·(a3·U_t + a5·U_tdt + a7·U_2dt)  Вычисление вектора эффективной нагрузки
  U_new ← Isolve(K_hat,R_eff)  Вычисление Ut+Δt
  U_2dt ← U_tdt  Вычисление Ut-2Δt
  U_tdt ← U_t  Вычисление Ut-Δt
  U_t ← U_new  Вычисление Ut
  T ← stack[T, (i·dt U_new_0 U_new_1)]  Формирование таблицы
T

```

Рис. 1. Код функции реализующий расчет методом Хаболта.

Контрольный пример:

$$\underline{\underline{K}} := \begin{pmatrix} 6 & -2 \\ -2 & 4 \end{pmatrix} \quad \underline{\underline{M}} := \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \quad \underline{\underline{R_tdt}} := \begin{pmatrix} 0 \\ 10 \end{pmatrix} \quad \underline{\underline{U_2dt}} := \begin{pmatrix} 0.0 \\ 0.392 \end{pmatrix} \quad \underline{\underline{U_dt}} := \begin{pmatrix} 0.0307 \\ 1.45 \end{pmatrix} \quad \underline{\underline{U_t}} := \begin{pmatrix} 0.16 \\ 2.80 \end{pmatrix}$$

$$\underline{\underline{C}} := 0 \quad \underline{\underline{dt}} := 0.2 \varepsilon$$

$$\underline{\underline{T}} := \text{Habolit}(\underline{\underline{K}}, \underline{\underline{M}}, \underline{\underline{C}}, \underline{\underline{dt}}, \underline{\underline{R_tdt}}, \underline{\underline{U_t}}, \underline{\underline{U_dt}}, \underline{\underline{U_2dt}})$$

	0	1	2
0	"t"	"U1"	"U2"
1	0.28	0	0.392
2	0.56	0.031	1.45
3	0.84	0.16	2.8
4	1.12	0.446	4.083
5	1.4	0.901	5.009
6	1.68	1.479	5.416
7	1.96	2.081	5.289
8	2.24	2.578	4.745
9	2.52	2.843	3.983
10	2.8	2.792	3.218
11	3.08	2.402	2.621
12	3.36	1.728	2.279

Рис.2. Контрольный пример вычисления методом Хаболта.

Следует отметить, что пошаговая процедура, основанная на методе Хаболта, может быть применена в статических задачах, когда не учитываются инерционный эффект и демпфирование. Другими словами, если $C=0$ и $M=0$, процедура Хаболта дает решение квазистатической задачи для нагрузки, зависящей от времени.

Литература

1. Бате К., Вилсон Е. Численные методы анализа и метод конечных элементов /Пер. с англ. А.С. Алексеева и др.; Под ред. А.Ф. Смирнова. — М.: Стройиздат, 1982.