

Второй отборочный этап

Индивидуальная часть

В индивидуальной части Вам будет предложено 6 задач, по следующим тематикам:

1. Общая (25 баллов);
2. Физика ядерных реакторов (25 баллов);
3. Численное решение дифференциальных уравнений (25 баллов);
4. Программирование на языке С (10 баллов).
5. Программирование на языке С (5 баллов).
6. Программирование на языке С (10 баллов).

Задача II.1.1. Общая (25 баллов)

При пуске ядерного реактора важной является задача предсказания условий достижения реактором состояния критичности. Одним из используемых на практике методов является последовательное извлечение поглощающих стержней в подкритическом реакторе с фиксацией установившегося уровня нейтронного потока на каждой ступени подъема с последующей математической обработкой полученных данных и определения прогноза количества извлекаемых стержней, при котором реактор станет критическим.

Предлагается:

Определить по экспериментальным данным, приведенным ниже, количество поглощающих стержней, которые необходимо последовательно (индивидуально или группами) вывести (полностью вывести из активной зоны) из нижнего положения для достижения критичности реактора.

Начальные условия:

1. Реактор в стационарном состоянии. Глубина подкритичности: $\rho_0 = -3\beta$, $\beta = 0.0065$. Уровень нейтронной мощности: $n_0 = (1.0 \times 10^{-5})N_n$, N_n — номинальный уровень мощности.
2. Стержни аварийной защиты — взведены. Компенсирующие стержни и стержни регулирования — на нижних концевых выключателях (т. е. полностью погружены в активную зону)
3. Веса всех поглощающих стержней — одинаковые

При последовательном извлечении стержней получены следующие результаты измерения установившейся нейтронной мощности:

Число извлеченных стержней	1	5	10	20	25
Установившейся уровень мощности, n_i	1,034 n_0	1,2 n_0	1.5 n_0	3,0 n_0	6,0 n_0

Решение

Исходим из уравнения кинетики подкритического реактора в одногрупповом приближении при наличии внешнего источника нейтронов с интенсивностью S (без него невозможно управление подкритическим реактором):

$$\frac{dn(t)}{dt} = \frac{\rho - \beta}{\Lambda} n(t) + \lambda \cdot C(t) + S \quad (1)$$

$$\frac{dC(t)}{dt} = \frac{\beta}{\Lambda} n(t) - \lambda \cdot C(t) \quad (2)$$

Стационарное состояние определяется глубиной подкритичности $\rho_0 < 0$ и мощностью внешнего источника S :

$$\begin{cases} 0 = \frac{\rho_0 - \beta}{\Lambda} n_0 + \lambda C_0 + S, \\ 0 = \frac{\beta}{\Lambda} n_0 - \lambda C_0, \text{ откуда имеем (3)} \\ n_0 = \frac{\Lambda S}{-\rho_0} \end{cases}$$

Видно, что при приближении глубины подкритичности к нулю ($-\rho \rightarrow 0$), то есть к состоянию критичности реактора, стационарное (установившееся) значение мощности стремится к бесконечности, при этом $1/n \rightarrow 0$. Извлечение каждого стержня уменьшает по модулю глубину подкритичности на $\Delta\rho$, равную весу стержня, то есть $|\rho_i| = |\rho_i - 1| - |\Delta\rho|$, что приведет к установлению нового стационарного уровня нейтронной мощности n_i

$$n_i = \frac{\Lambda S}{-\rho_i}$$

По условию задачи, нам известны установившиеся уровни мощности подкритического реактора при извлечении различного числа стержней.

Построив график $1/n_{\text{уст}}$ от числа извлеченных стержней и продлив его до пересечения с осью абсцисс (что будет соответствовать $1/n_{\text{уст}} = 0$, или $n_{\text{уст}} = \infty$), определяем число извлеченных стержней, при котором достигается критичность.

Для условий задачи из графика получаем, что искомое число стержней равно 30.

Задачу можно решить другим способом, определив вес стержня $|\Delta\rho|$ из данных по изменению стационарного уровня мощности при извлечении известного количества стержней, исходя из соотношения (получим $|\Delta\rho| = 0,1\beta$). После этого искомое число извлекаемых стержней, при котором достигается критичность, можно определить, разделив величину исходной подкритичности ($\rho_0 = -3\beta$) на вес стержня:

$$\left| \frac{\rho_0}{\Delta\rho} \right| = 30$$

Ответ: 30.

Изучение свойства саморегулирования энергетических реакторов при изменениях параметров активной зоны, приводящих к изменению реактивности. Среди этих параметров — изменения мощности, температуры топлива, теплоносителя, замедлителя и т. д. в физике реакторов эти явления называют эффектами реактивности, которые в статике характеризуют коэффициентами реактивности. Естественно, для обеспечения свойства саморегулирования необходимо, чтобы суммарный эффект (коэффициент) реактивности должен быть отрицательным, что должно быть обеспечено при проектировании реактора.

Для упрощения задачи часто рассматривают только мощностной коэффициент реактивности, аккумулирующий большинство частных факторов и который может быть легко определен экспериментально по результатам реализации режимов перехода с одного стационарного уровня мощности реактора на другой.

Задача II.1.2. Физика ядерных реакторов (25 баллов)

Определить значение мощностного коэффициента реактивности (в долях β) по переходному процессу изменения мощности до нового стационарного состояния путем внесения положительной реактивности заданным перемещением вверх стержня регулирования.

Исходные данные:

1. Начальный стационарный уровень нейтронной мощности: $0,5N_{\text{ном}}$.
2. Величина перемещения вверх стержня регулирования: $0,5$ м.
3. Высота активной зоны (длина хода стержня): 300 см.
4. Вес (эффективность) стержня: $0,2\beta$ (β — доля запаздывающих нейтронов).

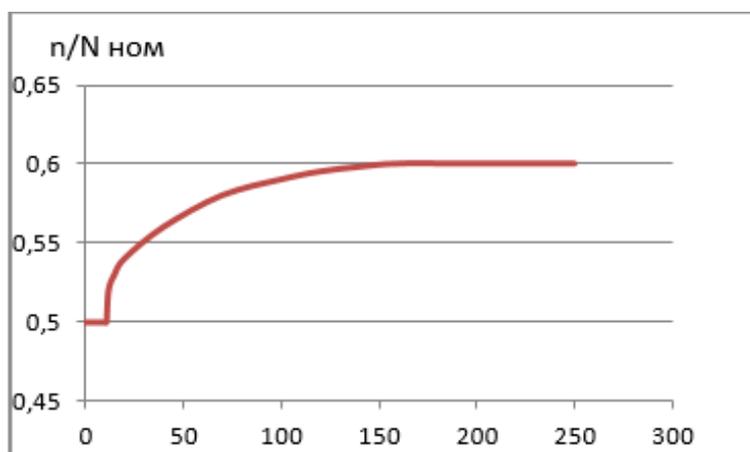


График переходного процесса изменения нейтронной мощности во времени (в секундах) из стационарного состояния при внесении положительной реактивности перемещением вверх регулирующего стержня.

В ответе указать значение мощностного коэффициента реактивности (в долях β) с точностью до сотых долей. В поле ввода указать только число.

Решение

Исходим из уравнения кинетики критического реактора в одногрупповом приближении в энергетическом диапазоне мощности:

$$\frac{dn(t)}{dt} = \frac{\rho - \beta}{\Lambda} n(t) + \lambda C(t), \quad (1)$$

$$\frac{dC(t)}{dt} = \frac{\beta}{\Lambda} n(t) - \lambda C(t), \quad (2)$$

$$T_M \frac{f \Delta \rho_M}{dt} + \Delta \rho_M = K_M \Delta n, \quad (3)$$

$$\rho(t) = \rho_{\text{вн}}(t) + \Delta \rho_M \quad (4)$$

$$\Delta n(t) = n_0 - n(t) \quad (5)$$

Уравнение (3) описывает в динамике изменение реактивности, вызываемое изменением нейтронной мощности реактора.

При реализации переходного режима, вызванного, например, ступенчатым изменением реактивности за счет перемещения стержня регулирования, реактор перейдет на новый стационарный уровень, так как внесенная реактивность будет скомпенсирована внутренней обратной связью по реактивности, определяемой мощностным эффектом. Таким образом, при достижении нового уровня мощности суммарная реактивность (см. уравнение (4)) станет нулевой (стационарному состоянию реактора соответствует нулевая реактивность). При этом уровень мощности изменится на величину, определяемую уравнением (3) в статике, то есть когда $\frac{d\Delta\rho_M}{dt} = 0$:

$$\Delta \rho_M = K_M \Delta n$$

При этом, как мы уже отметили, в статике $0 = \rho_{\text{вн}} + \Delta \rho_M$,

где $\rho_{\text{вн}}$ — положительное внешнее изменение реактивности, вызванное перемещением вверх стержня регулирования, равно:

$$\rho_{\text{вн}} = \frac{\text{вес стержня}}{(0,5 \text{ м}/3,0 \text{ м})} = 0,2\beta/6 = 0,033\beta \text{ (вес стержня приведен по модулю)}$$

При этом в статике $\Delta \rho_M = -\rho_{\text{вн}}$

Их графика, приведенного в задании, имеем $\Delta n = 0,1$ (в долях от $N_{\text{ном}}$) Отсюда:

$$K_M = \frac{\Delta \rho_M}{\Delta n} = \frac{-0,033\beta}{0,1} = -0,33\beta$$

Ответ: -0.33 ± 0.05 .

Задача II.1.3. Численное решение дифференциальных уравнений (25 баллов)

В некоторой системе протекает динамический процесс, заданный дифференциальным уравнением $\frac{dy}{dt} + 0.2 \cdot y(t) = 1$ при начальном условии $y(0) = 0$, где $y(t)$ — функция времени, определяющая сигнал на выходе системы.

Найти решение заданного уравнения с помощью построения разностной схемы методом Эйлера. Для расчета выбрать шаг по времени Δt , равный 1с .

Какого значения достигнет сигнал $y(t)$ на выходе системы в момент времени, равный 3с ?

Ответ дать с точностью до сотых долей.

Решение

Метод Эйлера является простейшим алгоритмом численного решения обыкновенных дифференциальных уравнений.

Пусть имеется дифференциальное уравнение $\frac{dy}{dt} = f(y, t)$, где f — некоторая функция. Для численного решения этого уравнения методом Эйлера задается размер шага по времени Δt . При этом алгоритм вычисления значений функции $y(t)$ в моменты времени, определяемые размером шага Δt , имеет вид

$$y_i = y_{i-1} + \Delta t \cdot f_{i-1} \quad (1)$$

где i — номер шага (итерации). Другими словами, запись y_i означает значение функции y в момент времени, равный $i \cdot \Delta t$, т. е. $y_i = y(i \cdot \Delta t)$.

По условию задачи динамический процесс задан дифференциальным уравнением $\frac{dy}{dt} + 0.2 \cdot y(t) = 1$, для которого можем записать: $\frac{dy}{dt} = 1 - 0.2 \cdot y(t)$.

Разностная схема решения этого уравнения методом Эйлера в соответствии с (1) будет иметь следующий вид:

$$y_i = y_{i-1} + \Delta t \cdot (1 - 0.2 \cdot y_{i-1}) \quad (2)$$

Таким образом, на первом шаге будем иметь $y_1 = y_0 + \Delta t \cdot (1 - 0.2 \cdot y_0)$ и, с учетом нулевого начального условия $y_0 = y(0) = 0$, получим $y_1 = \Delta t = 1$.

На втором шаге

$$y_2 = y_1 + \Delta t \cdot (1 - 0.2 \cdot y_1) = 1 + 1 \cdot (1 - 0.2) = 1.8$$

И, наконец, на третьем шаге, который соответствует моменту времени, равному 3с ,

$$y_3 = y_2 + \Delta t \cdot (1 - 0.2 \cdot y_2) = 1.8 + 1 \cdot (1 - 0.2 \cdot 1.8) = 2.44.$$

Ответ: $2,44 \pm 0.05$.

Задача II.1.4. Программирование на языке C (10 баллов)

Найдите ошибку в коде программы. Дайте ответ в виде строки, содержащей ошибку.

```

1 #include <stdio.h>
2 #define n 2
3 double get_double(void);
4 struct complex_number

```

```

5  {
6      double x;
7      double y;
8  }cn[n];
9  int main()
10 {
11     int k = 0;
12     double Qx = 0, Qy = 0, Qx1 = 0, Qy1 = 0;
13     char t;
14     for (k = 0; k < 3; k++)
15     {
16         if (k == 0)
17             t = 'a';
18         else if (k == 1)
19             t = 'b';
20         else t = 'c';
21         printf("\nEnter Complex number %c \n", t);
22         printf("Real x = ");
23         cn[k].x = get_double();
24         printf("Image y = ");
25         cn[k].y = get_double();
26     }
27     printf("Let's make an operation Q = a * (b - c) - c^2:\n");
28     Qx = cn[1].x - cn[2].x;
29     Qy = cn[1].y - cn[2].y;
30     Qx1 = Qx;
31     Qy1 = Qy;
32     Qx = cn[0].x * Qx1 - cn[0].y * Qy1;
33     Qy = cn[0].y * Qx1 + cn[0].x * Qy1;
34     Qx1 = Qx;
35     Qy1 = Qy;
36     Qx = Qx1 - cn[2].x * cn[2].x + cn[2].y * cn[2].y;
37     Qy = Qy1 - cn[2].x * cn[2].y - cn[2].y * cn[2].x;
38     printf("Q = %f + i * %f \n", Qx, Qy);
39     return 0;
40 }
41 double get_double(void)
42 {
43     double input;
44     char ch;
45     while (scanf_s("%lf", &input) != 1)
46     {
47         while ((ch = getchar()) != '\n')
48             putchar(ch);
49         printf("ERROR!!!\n");
50     }
51     return input;
52 }

```

Решение

В данной задаче проводится работа с массивом структур. Индексация структур начинается с 0. При описании шаблона

```

1  struct complex_number
2      {
3      double x;
4      double y;

```

```

5     }
6     cn[n];

```

задается массив структур. Директива препроцессора `#define n 2` задает значение параметра `n`, равное 2. Но, в таком случае, массив состоит из 2 структур. А в коде программы идет обработка 3 структур. Поэтому необходимо исправить строку `#define n 2` на `#define n 3`.

Ответ: `#define n 2`.

Задача II.1.5. Программирование на языке C (5 баллов)

Какое сообщение будет выдано в результате выполнения программы?

```

1  #include <stdio.h>
2  long function(int n)
3  {
4      return ((n == 1) ? 1 : n * function(n - 1));
5  }
6
7  int main()
8  {
9      int x = 5, y;
10     y = function(x);
11     printf("Result is %d \n", y);
12 }

```

Решение

В этой задаче используется рекурсивный вызов функции (т. е. функция вызывает саму себя). С каждым новым вызовом функции параметр `n` уменьшается на 1. Когда значение параметра `n` станет равным 1, функция возвратит 1.

Предположим, `n=5`

Получаем результат:

- на первом вызове функции `5*function(4)`
- на втором вызове функции `5*4*function(3)`

и т. д. Нетрудно заметить, что данная функция позволяет рассчитать факториал числа. Ответом будет строка `Result is 120` (факториал числа 5).

Ответ: `Result is 120`.

Задача II.1.6. Программирование на языке C (10 баллов)

Напишите программу на языке C, реализующую рекуррентное соотношение $y_i = y_{i-1} + T \cdot (1 - 2 \cdot y_{i-1})$ при начальном условии $y_0 = 1$, где T — параметр алгоритма, равный 0.1.

Выводом программы должно быть значение, к которому стремится значение переменной y ?

Решение

Один из вариантов решения задачи приведен ниже:

```
1  #include <stdio.h>
2  void main()
3  {
4      float y=1, T=0.1;
5      for(;;)
6      {
7          y=y+T*(1-2*y);
8          printf("%f\n",y);
9      }
10 }
```

Используя бесконечный цикл, можно увидеть, что значение переменной стремится к 0.5.

Командная часть

Тема: изменение нейтронной мощности критического реактора на малых уровнях мощности при внесении положительной реактивности.

Начальные условия и исходные данные:

1. Использовать модель кинетики точечного реактора нулевой мощности с одной эквивалентной группой запаздывающих нейтронов.
2. Параметры модели:

$n(t)$	плотность нейтронов, в долях от начального стационарного уровня n_0
$C(t)$	концентрация ядер-эмиттеров запаздывающих нейтронов
β	доля запаздывающих нейтронов ($\beta = 0,0065$)
ρ	реактивность реактора
Λ	время генерации (жизни) мгновенных нейтронов ($\Lambda = 10^{-3}c$)
λ	интенсивность распада ядер-эмиттеров запаздывающих нейтронов ($\lambda = 0, 1c^{-1}$)

Начальное состояние — стационарное, $\rho = 0$, $n(0) = 1$

Входное воздействие: $\Delta\rho = 0,05\beta$

Задача II.2.1. Командная задача (100 баллов)

Найти решение уравнения кинетики реактора «нулевой» мощности при скачкообразном внесении небольшой положительной реактивности. Определить значения относительного изменения нейтронной мощности в заданные моменты времени от начала переходного процесса (рассчитать $n(t)$ в моменты времени: $t = 2c, 10c, 50c, 100c$).

Этапы выполнения задания:

1. Используя методический материал, подготовить систему дифференциальных уравнений, описывающих кинетику реактора.
2. Построить разностную схему для системы дифференциальных уравнений по методу Эйлера.
3. Реализовать полученный на основе разностной схемы алгоритм расчета переходного процесса в виде программы на языке С. Рекомендуется при этом выбирать значение расчетного шага по времени Δt не более 0,1 с.
4. С помощью полученной программы рассчитать значения относительного изменения нейтронной мощности (плотность нейтронов в долях от начального стационарного уровня) в заданные моменты времени.
5. По результатам выполнения заданий подготовить краткий отчет.

Численный ответ дать в виде полученных значений, округленных до сотых долей, разделенных знаком «пробел».

Например: 3.33 4.44 5.55 6.66

Решение

Система линейных дифференциальных уравнений, описывающая временное поведение плотности нейтронов в «точечном» приближении при изменении реактивности в реакторе с одной эффективной группой запаздывающих нейтронов в отсутствие внешнего источника, имеет вид:

$$\begin{cases} \frac{dn(t)}{dt} = \frac{\rho - \beta}{\Lambda} n(t) + \lambda \cdot C(t), \\ \frac{dC(t)}{dt} = \frac{\beta}{\Lambda} n(t) - \lambda \cdot C(t) \end{cases} \quad (1.1)$$

Известно, что начальное состояние (до введения реактивности) — стационарное. Это означает, что производные $\frac{dn(t)}{dt}$ и $\frac{dC(t)}{dt}$ равны нулю. Тогда получаем систему уравнений, определяющую стационарное состояние

$$\begin{cases} \frac{\rho - \beta}{\Lambda} n(0) + \lambda \cdot C(0), \\ \frac{\beta}{\Lambda} n(0) - \lambda \cdot C(0) = 0 \end{cases} \quad (1.2)$$

Учитывая также, что в начальном стационарном состоянии реактивность $\rho = 0$, из (1.2) находим начальное значение для концентрации ядер-эмиттеров запаздывающих нейтронов $C(0) = \frac{\beta}{\lambda \Lambda} n(0) = \frac{\beta}{\lambda \Lambda} = 65$.

Таким образом, для расчета плотности нейтронов $n(t)$ необходимо решать систему дифференциальных уравнений (1.1) при начальных условиях $n(0) = 1$ и $C(0) = \frac{\beta}{\lambda \Lambda} = 65$.

2. Метод Эйлера является простейшим алгоритмом численного решения обыкновенных дифференциальных уравнений.

Пусть имеется дифференциальное уравнение $\frac{dy}{dt} = f(y, t)$, где f — некоторая функция. Для численного решения этого уравнения методом Эйлера задается размер шага по времени Δt . При этом алгоритм вычисления значений функции $y(t)$ в моменты времени, определяемые размером шага Δt , имеет вид

$$y_i = y_{i-1} + \Delta t \cdot f_{i-1}, \quad (2.1)$$

где i — номер шага (итерации). Другими словами, запись y_i означает значение функции y в момент времени, равный $i \cdot \Delta t$, т. е. $y_i = y(i \cdot \Delta t)$. Для отыскания решения

системы дифференциальных уравнений (1.1) методом Эйлера, необходимо построить разностную схему в соответствии с (2.1) для каждого уравнения системы (1.1):

$$n_i = n_{i-1} + \Delta t \cdot \left(\frac{\rho - \beta}{\Lambda} n_{i-1} + \lambda \cdot C_{i-1} \right),$$

$$C_i = C_{i-1} + \Delta t \cdot \left(\frac{\beta}{\Lambda} n_{i-1} - \lambda \cdot C_{i-1} \right).$$

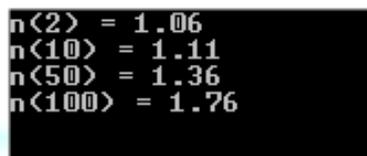
3. Для реализации полученного алгоритма можно предложить, например следующую программу на языке C:

```

1  int main()
2  {
3      //Задание параметров
4      float beta = 0.0065; //доля запаздывающих нейтронов
5      float ro = 0.05*beta; //заданная вносимая реактивность
6      float L = 0.001; //время генерации (жизни) мгновенных нейтронов
7      float l = 0.1; //интенсивность распада ядер-эмиттеров запаздывающих нейтронов
8      float n = 1.0; //начальное значение плотности нейтронов
9      float C = 65.0; //начальное значение концентрации ядер-эмиттеров запаздывающих
    ↪ нейтронов
10     int N = 1000; //количество шагов (итераций)
11     float T = 0.1; //размер шага по времени
12     float t = 0; //начальное значение времени
13     float n_[2], C_[2]; //массивы для хранения предыдущего (i-1) и текущего (i)
    ↪ значений плотности нейтронов и концентрации ядер-эмиттеров запаздывающих
    ↪ нейтронов
14     n_[0] = n; C_[0] = C; //инициализация начальных значений плотности нейтронов и
    ↪ концентрации ядер-эмиттеров запаздывающих нейтронов
15
16     //Реализация разностной схемы
17     for (int i = 1; i <= N; i++)
18     {
19         n_[1] = n_[0] + T*((ro - beta)*n_[0] / L + l*C_[0]); //расчет значения
    ↪ плотности нейтронов на i-м шаге
20         C_[1] = C_[0] + T*(beta*n_[0] / L - l*C_[0]); //расчет концентрации
    ↪ ядер-эмиттеров запаздывающих нейтронов на i-м шаге
21         t = i * T; //текущее значение времени
22         if (t == 2.0 || t == 10.0 || t == 50.0 || t == 100.0) //вывод
    ↪ результатов расчета в заданные моменты времени с округлением до
    ↪ сотых долей
23         {
24             printf("n(%d) = %0.2f\n", (int)t, n_[1]);
25         }
26         n_[0] = n_[1];
27         C_[0] = C_[1];
28     }
29     return 0;
30 }

```

Результат выполнения программы показан на рисунке:



```

n(2) = 1.06
n(10) = 1.11
n(50) = 1.36
n(100) = 1.76

```

Ответ: 1.06 1.11 1.36 1.76.