в виде равенств для целевой функции. В качестве ограничений типа неравенств в настоящей задаче использованы нелинейные условия прочности Хилла [3], которые являются условием прочности для внешних слоев пластины. Для заполнителя использованы ограничения типа:

$$|\tau_{max}| \le R_{T2},$$

где R_{T2} – расчетное сопротивление материала заполнителя сдвигу.

Расчет гофрированных конструкций сопряжен со значительными трудностями, которые возникают из-за сложной метрики их срединной поверхности. Поэтому при оптимизации трехслойной пластины расчет упрощался путем замены гофрированного элемента пластины эквивалентным ему гладким. Нахождение параметров эквивалентного элемента пластины выполнялось посредством приравнивания изгибных жесткостей гофрированного и эквивалентного ему элемента

$$EI = EI^{экв}.$$

Для сжатых полок трапецеидального гофра обшивок пластины вводились ограничения по местной устойчивости $\sigma \le \sigma_{кр}$. Полки гофров рассматривались как удлиненные, шарнирно опертые, прямоугольные пластины на упругом основании, которым служит средний слой пластины, имеющий коэффициент постели [4]

$$k = E_3/H_п,$$

где E_3 – модуль упругости заполнителя; $H_п$ – толщина подкрепляющего слоя.

При расчете на местную устойчивость полок трапецеидальных гофрированных обшивок критическая сила определялась по формуле [4]

$$P_{кр\,min} = 2\sqrt{kD}\left(\sqrt{1+\frac{\pi^4}{\gamma^4}}+\frac{\pi^2}{\gamma^2}\right), \qquad (3)$$

где $\gamma = a\cdot\sqrt[3]{\frac{k}{D}}$; $D = \frac{\overline{E_{пр}}}{12}\delta$; $\overline{E_{пр}} = \frac{\overline{E}}{1-\overline{\mu}^2}$.

Здесь $\overline{E}$, $\overline{\mu}$ – модуль упругости и коэффициент Пуассона материала обшивки; D – цилиндрическая жесткость пластины. Подставляя в (3) выражения для входящих параметров и учитывая, что $\sigma_{кр} = P_{кр}/\delta$ после преобразования получили выражение критических напряжений:

$$\sigma_{кр} = \sqrt{\frac{\pi^4 \overline{E}_{пр}\delta^3 H_п}{a^4 E_3 12}}.$$

Кроме выше перечисленных ограничений условия технологической осуществимости и теплотехнического расчета требовали выполнения ограничения по минимальной толщине среднего слоя. Выборочные результаты расчетов приведены в таблице.

Нагрузка q, кН/м^2	Ширина полки, см			Высота гофров, см			Min толщина среднего слоя H_0, см	Стоимость материала пластины, $
	a_1	a_2	a_3	h_1	h_2	h_3		
1	2	3	4	5	6	7	8	9
δ = 0,1 см, n = 15 см								
2,5	3,09	3,27	3,07	3,97	1,91	1,44	2,43	109,3
3,0	3,06	3,55	3,86	3,83	1,92	1,18	3,62	117,5
4,0	3,34	4,32	4,32	3,8	1,64	0,84	5,52	130,2
δ = 0,1 см, n = 20 см								
2,5	3,07	3,39	3,32	3,97	1,93	1,44	2,69	115,1
3,0	2,92	3,30	3,45	3,97	1,90	1,43	3,65	121,7
4,0	2,90	4,40	3,87	3,89	1,73	1,29	5,31	130,0

Результаты оптимизации трехслойной пластины, при замене гофрированных элементов на эквивалентные им гладкие, позволяют снизить стоимость пластин на 15–18 % и получить оптимальные параметры гофров и стоимости пластины при различных загружениях.

Список литературы

1. Райзер В.Д., Должиков В.Н., Должикова Е.Н. Определение оптимальных параметров составных пластин методом нелинейного программирования // Строительная механика и расчет сооружений. – 1987. – № 1. – С. 21–23.

2. Должикова Е.Н. Оптимизация параметров ортотропных составных пластин: дис. … канд. техн. наук. – М.: МИСИ,1986. – 162 с.

3. Хилл Р. Математическая теория пластичности. – М.: Гостехиздат, 1956. – 407 с.

4. Ермолов С.Б. Устойчивость пластинок на упругом основании и элементов гофрированных обшивок трехслойных панелей // Расчет конструкций с применением пластмасс: Тр. ЦНИИСК им. В.А. Кучеренко. – 1974. – С. 101–108.

C++ ДЛЯ ГЕОДЕЗИСТОВ И КАРТОГРАФОВ. УЧЕБНАЯ ПРОГРАММА «КОЛЛИМАЦИОННАЯ ПОГРЕШНОСТЬ» С УСЛОВНОЙ IF-ELSE ИНСТРУКЦИЕЙ

Заблоцкий В.Р.

Московский государственный университет геодезии и картографии, Москва, e-mail: v-r-zablotskii@ya.ru

Обсуждается учебная программа на языке C++ для студентов, обучающихся программированию в вузе геодезического профиля. Нашей целью является создание набора типовых учебных геодезических задач [1, 2], которые могут использоваться студентами геодезистами и кар-

тографами в качестве домашних заданий и при выполнении учебного практикума по информатике. Задача данной работы заключалась в разработке программы, использующей управляющую инструкцию *if – else,* для демонстрации выбора между двумя ветвями программы.

Рассмотрим содержательную постановку задачи. Известно, что коллимационная погрешность теодолита связана с неперпендикулярностью оси визирования и оси вращения трубы прибора в горизонтальной плоскости. Для вычисления коллимационной погрешности теодолита используется формула: $2c$ = КЛ – КП ± 180°, где $2c$ – удвоенная погрешность, КЛ – отсчет, взятый с микроскопа при круге лево, КП – отсчет с микроскопа при круге право. Формула включает в себя два возможных случая. Если разность отсчетов при КЛ и КП больше нуля, то из полученной разности вычитается 180 градусов. Если же разность отсчетов при КЛ и КП меньше нуля, то к полученной разности прибавляется 180 градусов. Данная формула позволяет применить условную инструкцию *if – else,* в которой ветвь *if* вычисляет коллимационную погрешность со сложением 180°, а ветвь *else* с вычитанием 180°.

В программе «КОЛЛИМАЦИОННАЯ ОШИБКА» иллюстрируется последовательность действий для вычисления погрешности по результатам отсчетов КЛ и КП, полученных при визировании на некоторую точку.

```
01:  #include <iostream>
02:  using namespace std;
03:
04:  int main (void)
05:  {
06:   int readingInDegreesWithCL, readingInDegreesWithCR;
07:   float readingInMinutesWithCL, readingInMinutesWithCR;
08:   int totalSecondsCL, totalSecondsCR;
09:   float differenceOfReadings, collimationError;
10:
11:   cout<<"Введите отсчет КЛ, через пробел градусы и минуты: ";
12:   cin >>  readingInDegreesWithCL >> readingInMinutesWithCL;
13:
14:   cout<<"Введите отсчет КП, через пробел градусы и минуты: ";
15:   cin >>  readingInDegreesWithCR >> readingInMinutesWithCR;
16:
17:   totalSecondsCL = readingInDegreesWithCL * 3600 +
17:                                 readingInMinutesWithCL * 60;
18:
19:   totalSecondsCR = readingInDegreesWithCR * 3600 +
19:                                 readingInMinutesWithCR * 60;
20:
21:   differenceOfReadings = totalSecondsCL - totalSecondsCR;
22:
23:   if (differenceOfReadings > 0)
24:     collimationError = (differenceOfReadings - 180*3600)/2;
25:   else
26:     collimationError = (differenceOfReadings + 180*3600)/2;
27:
28:   if(collimationError < 0)
29:     collimationError = -collimationError;
30:
31:   cout <<"Коллимационная погрешность теодолита 2Т30: "
32:        << collimationError <<"\" "<< endl;
33:
34:   if (collimationError > 60)
35:   {
36:     cout <<"Погрешность не должна превышать двойной точности "
37:          <<"инструмента. Требуется юстировка 2Т30." << endl;
38:   }
39:   else
40:   {
41:     cout << "Погрешность трубы теодолита в допуске. "
42:          << "Переходите к следующей поверке." << endl;
43:   }
44:   return 0;
45:  }
```

Предположим, что были введены следующие данные: отсчет при КЛ равен 0°0.5′ и отсчет при КП равен 180°2′. Программа на экране дисплея напечатает следующее: «Коллимационная погрешность: 45″. Погрешность трубы теодолита 2Т30 в допуске. Переходите к следующей поверке». Если были введены другие данные, например: отсчет при КЛ равен 23°18.5′ и отсчет при КП равен 203°16′, тогда программа на экране напечатает следующее: «Коллимационная погрешность: 75″. Погрешность не должна превышать двойной точности инструмента. Требуется юстировка теодолита 2Т30».

Обращаем внимание, что некоторые строки в коде программы, имеют одинаковую нумерацию, поскольку длинная строка кода из-за недостатка места была искусственно разбита на две отдельные части. В строках 06–07 объявлены переменные: отсчет в градусах при КЛ *readingInDegreesWithCL* и в минутах *readingInMinutesWithCL*, отсчет в градусах при КП *readingInDegreesWithCR* и в минутах *readingInMinutesWithCR*. Объявлены также переменные для хранения отсчетов при КЛ и КП в угловых секундах *totalSecondsCL* и *totalSecondsCR*, а также переменная для разности отсчетов при КЛ и КП *differenceOfReadings* и коллимационная погрешность *collimationError*. Ввод данных выполняется в строках 12 и 15, затем величина отсчета пересчитывается в угловые секунды, для этого используются переменные *totalSecondsCL* и *totalSecondsCR*, соответственно для КЛ и КП. Далее вычисляется коллимационная погрешность трубы теодолита по формуле, представленной в строке 24, если *differenceOfReadings > 0*, в противном случае по формуле в строке 26. В строках 23–26 используется условная инструкция в полном формате, то есть имеющем две ветви альтернативного выбора. Получившаяся коллимационная погрешность может иметь знак минус, поэтому используется дополнительная условная инструкция *if* в строках 28 и 29, чтобы погрешность, выводимая на экран, была положительной величиной. Данная условная инструкция записана в сокращенном формате (без ветки *else*) и если условие, стоящее в круглых скобках истинно, то эта инструкция не выполняется вовсе. Затем используется еще одна условная инструкция *if-else* в строках 34–43, резюмирующая полученный пользователем результат. Если коллимационная погрешность теодолита 2Т30 превышает 60 угловых секунд, на экране печатается текст о том, что требуется юстировка прибора.

Выводы. Разработана учебная программа на языке С++ для студентов, обучающихся программированию в вузе геодезического профиля. В программе демонстрируется применение управляющей инструкции *if – else* в задаче вычисления коллимационной погрешности теодолита. Данная программа подчеркивает особенности использования условной инструкции в полной и сокращенной форме для осуществления выбора действий на основе истинности или ложности некоторых условий.

Список литературы

1. Заблоцкий В.Р. Программирование на языке С++ для картографов и геодезистов: учебная объектно-ориентированная программа «Нивелирная рейка» // Международный журнал экспериментального образования. – 2015. – № 5 (часть 1). – С. 89–91.

2. Заблоцкий В.Р. С++ для картографов и геодезистов: учебная объектно-ориентированная программа «Женевская линейка» // Международный журнал экспериментального образования. – 2015. – № 10 (часть 1). – С. 25–26.

Физико-математические науки

СРЕДСТВА АВТОМАТИЗАЦИИ НЕСТАЦИОНАРНОГО АЭРОДИНАМИЧЕСКОГО ЭКСПЕРИМЕНТА

[1]Башуров В.В., [1,2]Гилев В.М., [2]Саленко С.Д., [2]Слободской И.В., [1]Шпак С.И.

[1]Институт теоретической и прикладной механики им. С.А. Христиановича СО РАН;
[2]Новосибирский государственный технический университет, Новосибирск, e-mail: gil@itam.nsc.ru

Представлены аппаратно-программные средства автоматизации нестационарного аэродинамического эксперимента. С помощью данных средств обеспечивается сбор экспериментальных данных, а также управление отдельными элементами аэродинамической трубы.

Введение. В представляемой работе описывается система автоматизированного управления и сбора информации аэродинамического эксперимента. В качестве объекта управления используется аэродинамическая труба дозвуковых скоростей Т-503 кафедры аэрогидродинамики (АГД) НГТУ с открытой рабочей частью.

В Новосибирском государственном техническом университете (НГТУ) проводятся фундаментальные и прикладные научные исследования в области нестационарной аэродинамики. Для выполнения подобных исследований, а также обучения студентов, магистрантов НГТУ основам аэродинамики и гидромеханики на кафедре аэрогидродинамики университета используется аэродинамическая труба дозвуковых скоростей Т-503 [1]. Данная установка является аэродинамической трубой замкнутого типа с открытой рабочей частью и имеет следующие основные характеристики:

– рабочий диапазон скоростей от 0 до 60 м/с;

– диаметр рабочей части 1,2 м, длина 2 м;

– неравномерность скорости в ядре потока диаметром 0,8 м не более 0,8 %;

– степень турбулентности без турбулизирующих устройств около 0,3 %.