

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
БЕЛАРУСЬ**

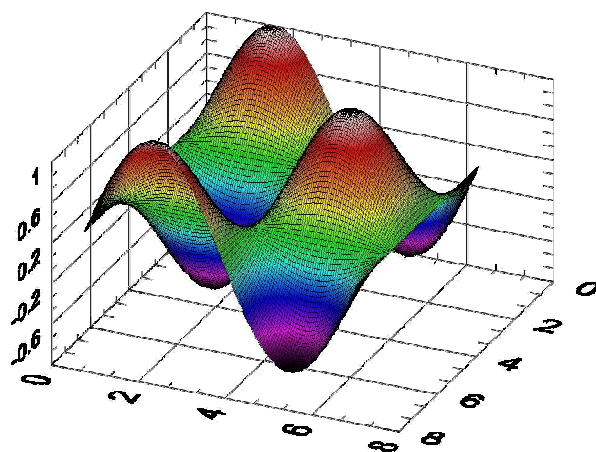
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТРАНСПОРТА»**

**Кафедра
«Микропроцессорная техника и информационно-управляющие
системы»**

М. В. НАФТОЛЬСКИЙ, В. В. ШЕВЦОВ

СПЕЦИЗМЕРЕНИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА

Лабораторный практикум



Гомель 2016

ЛАБОРАТОРНАЯ РАБОТА № 3

МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ В ИНЖЕНЕРНОЙ СРЕДЕ LabVIEW

Цель: изучение возможностей среды LabVIEW для математического моделирования физических явлений и процессов, создания подпрограмм обработки данных.

Задание. Разработать программу моделирования политропного процесса сжатия воздуха в цилиндре объемом $V_0=1$ л с начальным давлением $P_0=100$ кПа и температурой $T_0=300$ К при степени сжатия $\lambda=V_0/V_K=5$. Результаты вычислений отобразить в виде индикаторов традиционных приборов, служащих для измерения параметров состояния V , P , T , графиков их изменения по времени и P - V диаграммы исследованного процесса.

Основные положения

В настоящее время решение большинства научно-технических задач базируется на использовании компьютерного моделирования, когда этапам проведения физических экспериментов, разработки новых изделий и технологий предшествуют сложные математические расчеты, создание имитационных моделей и т.п.

В качестве примера в настоящей работе рассматриваются возможности инженерной среды графического программирования для моделирования термодинамических процессов идеального газа. Процессом называется любое изменение параметров его состояния. Обычно изменяются все три параметра, связанные между собой уравнением $PV=RT$. Существует ряд процессов, в течение которых сохраняется постоянное отношение выполненной работы и количества тепла, участвующего в теплообмене с внешней средой. Такие процессы называются политропными. Для них выполняется дополнительное соотношение $PV^n=const$.

Если в политропном процессе воздух, являющийся идеальным газом, сжимается очень быстро, то при уменьшении объема в 15 раз, температура его повышается до 650 °С. В него можно впрыснуть дизельное топливо, и оно самовоспламеняется.

Таким способом может быть реализован один из процессов, с помощью которого приводится в действие дизельный двигатель.

При той же степени сжатия, осуществляемого очень медленно, температура остается без изменений. Это связано с тем, что в медленном процессе тепловая энергия, которая образуется при сжатии газа, успевает рассеяться в окружающей среде. Таким образом, характер изменения параметров состояния фактически зависит от скорости процесса. В первом случае показатель политропы n равен коэффициенту адиабаты $n=1,4$ (адиабатический процесс), во втором $n=1$ (изотермический процесс).

Порядок выполнения задания

1.1. Осуществите запуск среды LabVIEW из каталога (например, D:\LABV). В появившемся главном окне программы выберите команды: *New* → *Blank VI* для создания нового файла. Далее выберите меню: *Window* → *Tile Left and Right* для одновременного отображения на экране двух окон программы - серой *Лицевой панели* и белой панели *Блок-диаграмм*. Для дальнейшей работы необходимо вызвать *Палитру инструментов* с помощью меню: *View* → *Show Tools Palette* на *Лицевой панели* или на *Блок-диаграмме*.

1.2. Создайте на *Лицевой панели* четыре цифровых элемента управления для исходных данных задачи: V_0 , P_0 , T_0 , n как показано на рисунке 3.1. Для этого щелчком ПКМ (правой кнопки мыши) по серой панели вызовите *Палитру элементов управления* (Controls) и закрепите ее, активизировав кнопку в левом верхнем углу палитры.

Откройте пункт меню *Num Ctrl*, выберите в нем первый элемент в верхнем ряду. На открывшейся *Палитре элементов управления* выделите элемент *Num Ctrl*. Переместите четыре элемента поочередно перетаскиванием на *Лицевую панель* и расположите их горизонтально в одну строку.

Измените собственные метки управляющих элементов, подписав вместо *Numeric* новые обозначения: V_0 , P_0 , T_0 , n . Установите в окошках регуляторов соответствующие значения исходных данных ($n=1$) с помощью инструмента «ввод текста (A)».

Для отображения полученных данных V, P, T создайте на *Лицевой панели* три прибора - мерную емкость, манометр и индикатор температуры. Подпишите названия этих приборов на русском языке и измените, верхние пределы их шкал - для объема - 1л, давления - 2000 кПа, температуры 1000 К. Для наблюдения за ходом процесса создайте на *Лицевой панели* трехлучевой запоминающий осциллограф и X-Y-самописец для построения P-V диаграммы процесса → *Палитры элементов управления* (Controls), графические индикаторы (Graph Inds), первый (Waveform Chart) и третий элемент (XY-Graph). Измените, легенды шкал, как показано на рисунке 3.1.

1.3 Освободите среднюю часть *Блок-диаграммы* для построения графического кода программы. Щелчком ПКМ на *Блок-диаграмме* вызовите *Палитру функций* (Functions→Express). Используя кнопку в верхнем левом углу палитры, зафиксируйте ее на экране.

В *Палитре Programming* вызвать первый элемент в первом ряду в виде квадрата с утолщенными сторонами, далее в нем выбираем цикл по условию *While*, перетаскиваем его на блок-диаграмму и растягиваем на большую часть экрана. Вернитесь к «Структурам», выберите формульный (Formula No...) узел и перенесите его внутрь цикла (рисунок 3.2).

В нашем случае определяющим процессом в моделировании является движение поршня и соответствующее изменение объема сжатого воздуха. Свяжем скорость этого процесса с показателем политропы следующим образом:

$$V = V_0 - 0,002 * k * n^5,$$

где $k = i$.

Таким образом, при моделировании время процесса принимается равным i - номеру текущей итерации цикла.

Это означает, что за каждый цикл моделирования процесса объем сжимаемого газа линейно уменьшается на $(2 * n^5)$ мл. Введем эту формулу в подготовленный узел и создадим на его границе три терминала ввода необходимых для расчета V данных - V0, k, n, а также один терминал для вывода V. Для этого щелчком ПКМ на границе узла вызываем всплывающее меню, в нем - три раза "Add Input" и один раз "Add Output".

Впишем в появившиеся терминалы буквенные обозначения переменных и с помощью инструмента «проводник» соединим их

соответственно с пиктограммами V_0 , n и узлом счетчика итераций, считая, что $k = i$.

1.4. Выведем полученное значение V на границу цикла, внесем в цикл пиктограмму мерной емкости, осциллограф и подадим значение V на их входы.

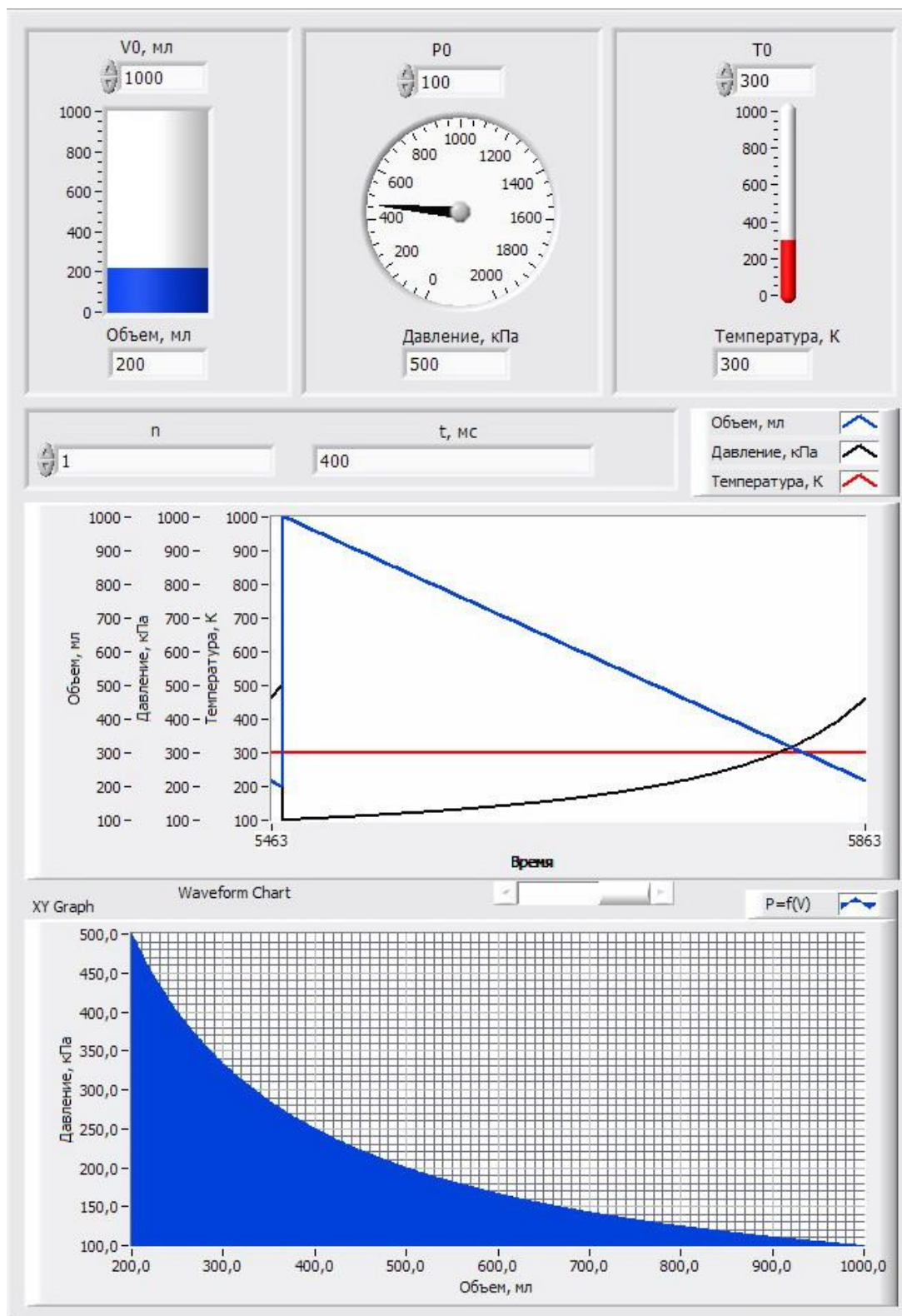


Рисунок 3.1 – Лицевая панель виртуального прибора для моделирования процессов сжатия

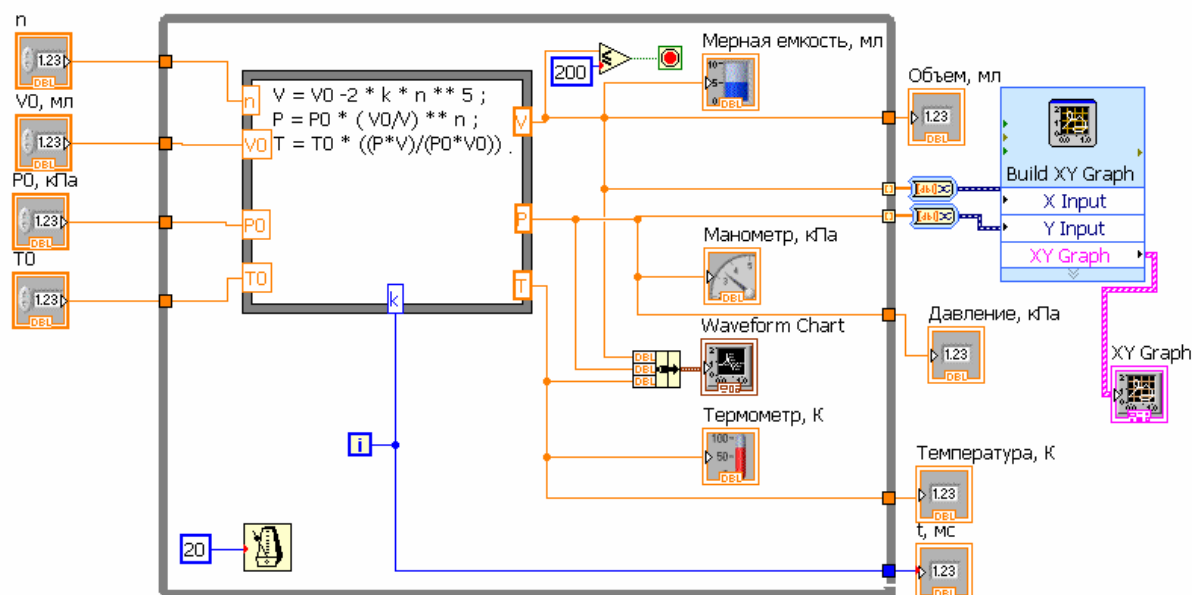


Рисунок 3.2 – Блок - диаграмма программы расчета параметров сжимаемого газа

Так как условие $\lambda = V_0 / V_k = 5$ (V_k меньше или равно 0,2 литра), используется для завершения цикла, находим в *Палитре Express* → *Programming* → *Comparison* → *Less Or Equal* (узел \leq) Подводим текущее значение V к верхнему терминалу логического узла, а к нижнему подводим константу 0,2. Логический результат *False* или *True* -присоединяем к терминалу завершения цикла. Обратим внимание на цвет проводника, соответствующий логическому типу данных. Так как по умолчанию каждый цикл рассчитывается всего за одну миллисекунду, то для отслеживания динамики процесса установим задержку цикла, равную 20 мс. Для этого на *Панели Express* → *Programming* → *Timing* → *Wait Until* выбираем пиктограмму прибора времени, в ней метроном. Помещаем его внутри цикла, находим входной терминал и щелчком ПКМ вызываем всплывающее меню, в нем *Create* → *Const.* В появившемся прямоугольнике с клавиатуры набираем число 20.

Убедитесь, что стрелка запуска цикла имеет правильную, не изломанную форму. Это означает, что программа составлена

правильно и готова к запуску. В противном случае - щелчком ПКМ по стрелке вызываем контекстное меню с распечаткой допущенных ошибок. Устраняем их и запускаем программу. При этом объем сжатого воздуха в мерной емкости за 10-15 секунд уменьшается от 1 до 0,2 литров, а на осциллографе появляется график уменьшения объема в виде прямой наклонной линии. На этом создание и отладка программного управления изменением объема сжимаемого газа завершается.

1.5 Соответствующее рассмотренному процессу изменение давления в цилиндре описывается формулой $P=P_0*(V_0/V)^n$.

Вносим это соотношение в формульный узел. Добавляем дополнительный вход для ввода P_0 и с помощью проводника связываем его с соответствующим элементом управления. На правой границе формульного узла создаем выход P и соединяем его с пиктограммой манометра, внесенной внутрь цикла.

Для одновременного отображения графиков изменения давления и объема воздуха на одном и том же приборе преобразуем однолучевой осциллограф в двухлучевой. Для этого в *Палитре Express* → *Programming* → *Cluster, Cla..* → *Bundle* выбираем массивы и кластеры, в них элемент «объединение». Активизируем проводник, соединяющий осциллограф с выходом V и убираем его. К нижнему входу элемента «объединение» подводим значение P , а у верхнего – восстанавливаем соединение с V . Выход элемента «объединение» соединяем с входом компьютерного осциллографа, который с этого момента становится двухлучевым.

Работа по созданию подпрограммы моделирования изменения давления при сжатии может считаться завершенной, если стрелка запуска имеет правильный вид. Для проверки правильности работы этой программы необходимо очистить предыдущий график щелчком ПКМ по экрану осциллографа набором команд во всплывающем меню *Data Operation* и *Clear Cap*.

Запустить программу и убедиться в том, что за время процесса давление в цилиндре изменяется от 100 до 500 кПа, а график его изменения по времени представляет собой возрастающую экспоненциальную функцию.

1.6 Изменение температуры в исследуемом процессе определяется соотношением: $T=T_0*((P*V)/(P_0*V_0))$. Введем его в формульный узел, добавим дополнительный вход T_0 и выход для полученного значения T . Соединим выход T и пиктограмму

термометра, помещенную внутри цикла. Добавим еще один канал соединения с осциллографом. Для этого с помощью курсора в виде стрелки активизируем элемент «объединение» и растянем его вниз на одну новую позицию. Подведем к образовавшемуся новому входу сигнал T и соединим общий выход с осциллографом.

Далее необходимо убедиться в правильности составленной подпрограммы, очистить прежний график и запустить программу целиком. При $n=1$ показания термометра остаются на том же уровне, так как этот изотермический процесс характеризуется как раз постоянным значением температуры. Давление и объем изменяются, как и в предыдущем случае.

1.7 Предусмотреть вывод конечных значений параметров моделируемого процесса на лицевую панель. Для этого щелчком ПКМ по серой панели вызвать 5 цифровых индикаторов и расположить их в следующей последовательности в соответствии с таблицей 3.1.

Таблица 3.1 – Цифровые индикаторы

n	t, мс	V _к , л	P _к , кПа	T _к , К
---	-------	--------------------	----------------------	--------------------

Вывести значения этих параметров на правую границу цикла и соединить их с соответствующими индикаторами. Повторить запуски программы при $n=1,2$ и $n=1,4$. Перенести измеренные значения в таблицу 3.2.

Таблица 3.2 – Результаты моделирования

№ п/п	Параметры				
	n	t, мс	V, л	P, кПа	T, К
1	1				
2	1,2				
3	1,4				

1.8 Полученные значения параметров состояния могут быть использованы для автоматического построения P-V диаграммы исследуемого процесса (рисунки 3.3 – 3.5). Для этого необходимо вывести на границу цикла текущее значение P и V. По умолчанию выходные терминалы в цикле While сохраняют только последние значения цикла как в пункте 1.7. Для того чтобы при моделировании были бы сохранены все значения, необходимо для

P и V изменить тип выходных данных. Для этого щелчком ПКМ по терминалу вызвать контекстное меню и поменять выходы (Enable Indexing). Далее соединить выходные терминалы V и P соответственно с X -Y входами двухкоординатного самописца. Обратит внимание, что толщина проводников для массивов чисел, передаваемых из выходных терминалов цикла, больше чем у проводников одиночных скалярных величин.

В очередной раз очистить графики и запустить программу. Проанализировать вид P-V диаграммы. Изменить форму представления данных на выделение области, лежащей под кривой P(V), и дать ее физическую интерпретацию

$$A = \sum P \cdot dv.$$

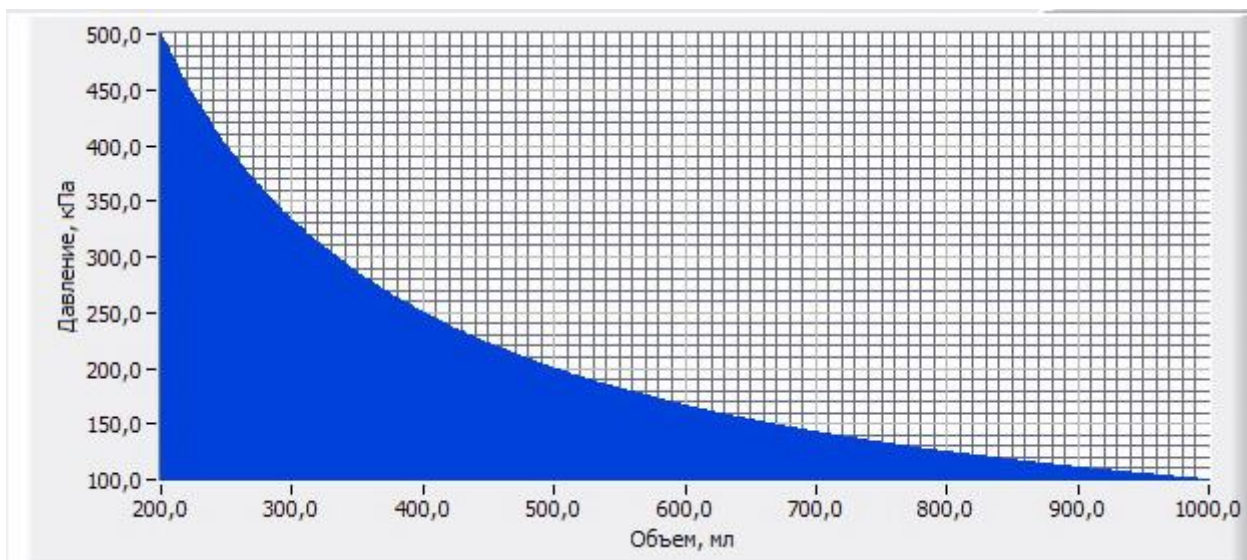


Рисунок 3.3 – P-V диаграмма при n = 1

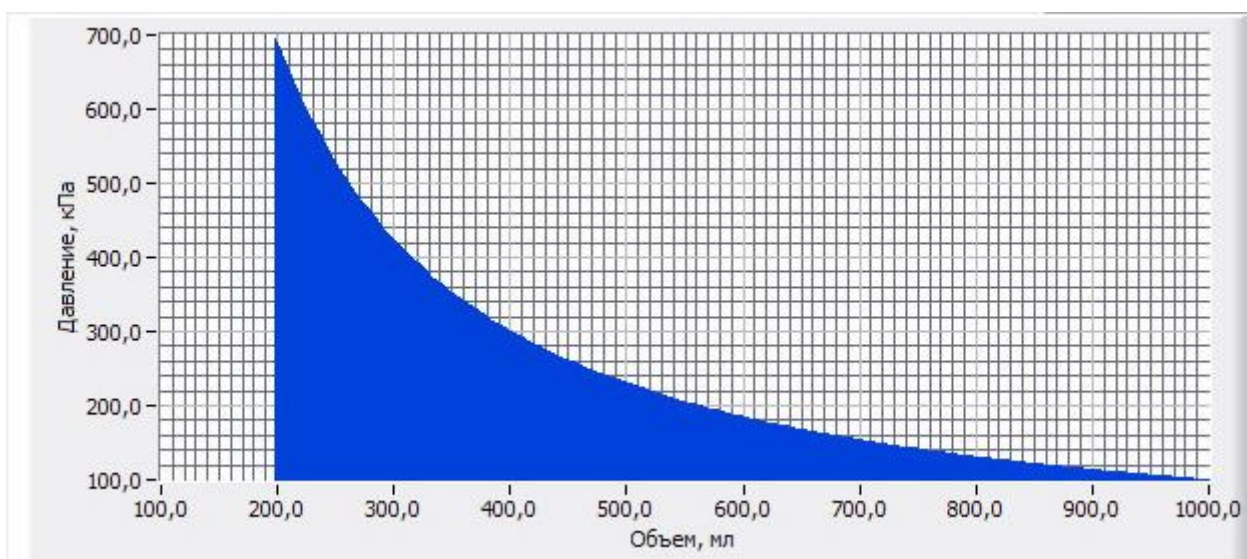


Рисунок 3. 4 – P-V диаграмма при $n = 1,2$

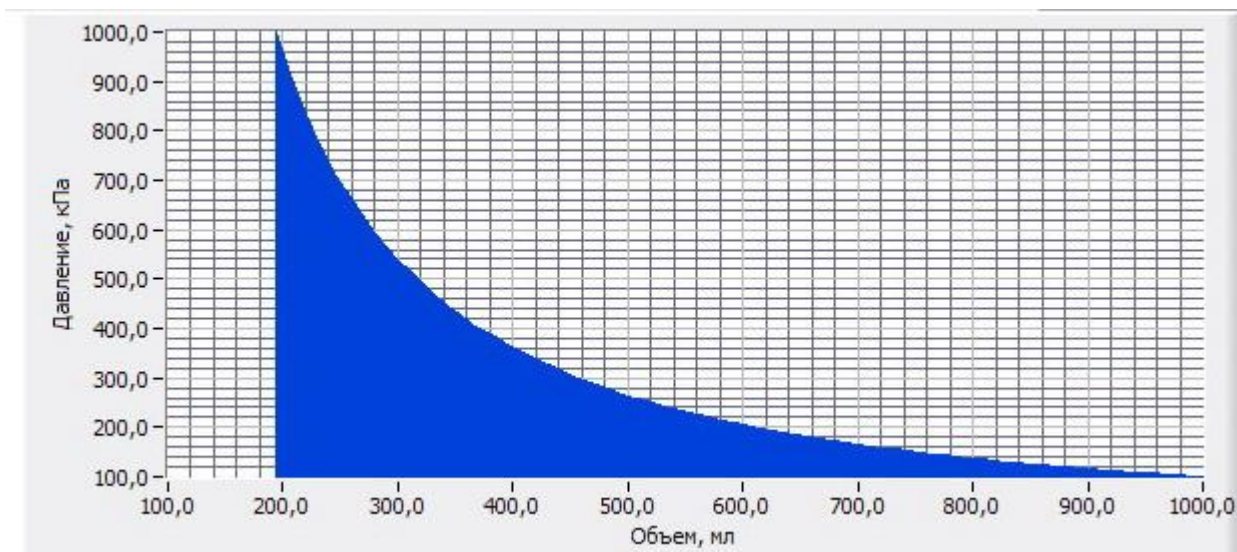


Рисунок 3.5 – P-V диаграмма при $n = 1,4$

1.9. Провести численное моделирование процесса при $n = 1$; 1,2 и 1,4. Полученные данные перенести в таблицу 3.2 и проанализировать их.

Требования к отчету

Наименование и цель работы; краткое описание термодинамических процессов идеального газа и процессов моделирования; разработанная программа моделирования политропного процесса сжатия воздуха в цилиндре (лицевая панель и блок-диаграмма); демонстрация работы программы в соответствии с заданием преподавателя; заключение.

Пример оформления выводов по работе

1. Разработана универсальная программа моделирования политропного процесса сжатия воздуха в цилиндре, объем, начальное давление и температура в котором могут быть заданы произвольным образом.

2. Установлено, что в медленном изотермическом процессе степень повышения давления равна степени сжатия воздуха. При быстром адиабатическом сжатии давление воздуха в несколько раз

превышает изотермическое и при степени сжатия 5 увеличивается более чем в 10 раз. Температура при этом достигает 560 К.

3. Построены P-V диаграммы исследуемых процессов. Показано, что площадь под кривой сжатия на P-V диаграмме численно равна механической работе затраченной на сжатие воздуха.

Контрольные вопросы

1. Приведите примеры других задач, которые могут решаться с помощью компьютерного моделирования.
2. Объясните, как работает цикл While.
3. Для чего на границах цикла формируются массивы данных P и V?
4. Что дает представление полученных зависимостей в виде площади под кривой процесса?
5. Назовите основные типы данных, использованных при выполнении задания.

СПИСОК ЛИТЕРАТУРЫ

1. Бутырин, П.А. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7 Express / П.А. Бутырин, Т.А. Васьковская, В.В. Каратаев, С.В. Материкин – М.: ДМК Пресс, 2005. - 264 с.

2. Суранов, С.Я. LabVIEW 8.5: справочник по функциям / С.Я. Суранов. - М.: ДМК Пресс, 2007. - 572 с.

3. Батоврин, В.К. LabVIEW: практикум по электронике и измерительной технике / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: ДМК Пресс, 2007.

4. Батоврин, В.К. LabVIEW: практикум по основам измерительных технологий / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: ДМК Пресс, 2007.

5. Евдокимов, Ю.К. LabVIEW для радиоинженера: от виртуальной модели до реального прибора / Ю.К. Евдокимов, В.Р. Линдваль, Г.И. Щербаков. - М.: ДМК Пресс, 2007.

6. Федосов, В.П. Цифровая обработка сигналов в LabVIEW / В.П. Федосов, А.К. Нестеренко. - М.: ДМК Пресс, 2007.

Некоторые замечания по технике отладки ВП в LabVIEW

Используйте Create Control/Constant так часто, как можете

В LabVIEW можно создавать константы на диаграмме простым щелчком по правой клавише мыши и выбором пункта *Create Constant*. Помните об этой возможности и со многими неприятными ошибками Вам не придется иметь дело. Пусть, например, в подпрограмме используется *Enumerate Constant* и для управления *Case* используется некоторое число, которое спустя некоторое время изменится, то Вы и не будете осведомлены о том, что случилось. Лучше создайте константу и тогда, если *enumerated constant* когда-нибудь изменится, Вы сразу получите сообщение об ошибке. Подумайте о часах, потраченных на поиск ошибок, которые может сохранить Вам этот прием.

Выполняйте соединение слева направо и не делайте соединения невидимыми

Выход установленный справа непосредственно и прямо соединяется с входом следующего блока.

Никогда не прокладывайте проводники один под другим! Если это сделать, Вы больше не сможете видеть выполнены все соединения или нет. К тому же Вы можете и забыть о том, чего не видите.

Могут возникнуть большие проблемы, если Вы делите некоторое количество точек в цикле и соедините их так, что изображение цикла будет накладываться на проводники. В этом случае проводники не будут видны, и Вы можете подумать, что эти проводники не участвуют в работе цикла.

Если Вы удаляете или перемещаете часть программы с невидимыми линиями, то будут удаляться и невидимые линии, а что будет после этого неизвестно. Если Вам повезет, то увидите предупреждение LabVIEW, если нет то...

Remove Broken Wires удобный инструмент, но

Избегайте использовать *Remove Bad Wires*. Этот инструмент удаляет любые неправильные соединения и не только те, на которые Вы смотрите, например те соединения, которые могут быть частью незаконченной программы. Если Вы удалите все, то потом нужно будет возвращаться назад и восстанавливать соединения в части программы, которую Вы не закончили. Некоторые соединения могут быть удалены, и при этом не будет вызова ошибки (равносильно удалением редактором всех строк с синтаксическими ошибками при программировании, например, на C).

Используйте *Remove Bad Wires* тогда, когда у Вас небольшая программа и возникло одно или два "плохих соединений". Вы абсолютно уверены, что ничего не испортите? Тогда нажимайте *Ctrl+B*.

Используйте тройной "клик" мышью для просмотра ошибки соединения

У вас возникли проблемы с соединением? Как их решить без применения Remove Bad Wires? Попробуйте щелкнуть мышью по плохому соединению. Одиночный щелчок выделит сегмент соединения, двойной от точки до точки, тройной всю линию. Тройной щелчок специально используют для поиска всех взаимосвязанных терминалов, тех о которых Вы знаете и не знаете. И нет другого пути, когда соединены два входных терминала (а этого нельзя делать) и один из проводников спрятан под иконкой. И если Вы будете использовать двойной щелчок и потом удалите все, то Вы удалите и правильную часть соединения. В такой ситуации следует трижды щелкнуть, чтобы увидеть все точки подключения. Нет более простого и правильного пути для того, чтобы быть уверенным в правильности соединения, чем тройной щелчок.