

**ПРИОРИТЕТНЫЙ НАЦИОНАЛЬНЫЙ ПРОЕКТ «ОБРАЗОВАНИЕ»
РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

Л.В. ВИНОГРАДОВ

**ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ
РАБОТЫ АВТОМОБИЛЬНЫХ ДИЗЕЛЕЙ
В УСЛОВИЯХ ВЫСОКОГОРЬЯ
И ЖАРКОГО КЛИМАТА**

Учебное пособие

Москва

2008

**«Создание комплекса инновационных образовательных программ
и формирование инновационной образовательной среды,
позволяющих эффективно реализовывать государственные интересы РФ
через систему экспорта образовательных услуг»**

Экспертное заключение –
доктор технических наук *В.И. Ерохов*

Виноградов Л.В.

Повышение эффективности работы автомобильных дизелей в условиях высокогорья и жаркого климата: Учеб. пособие. – М.: РУДН, 2008. – 187 с.

Учебно-методический комплекс включает девять тем.

Три темы рассматривают вопросы некоторых основ работы в интегрированной системе *Mathcad*.

Одна тема (четвертая) посвящена принципам модульного построения систем автоматизированного расчета.

Пять тем (шестая-девятая) касаются разработки программного обеспечения для расчета параметров дизеля при изменении условий окружающей среды и высокогорья. Программные модули могут быть применены непосредственно для расчетов, а также как программы-шаблоны для использования в других численных экспериментах.

Для магистров, обучающихся по направлению «Энергомашиностроение».

Учебное пособие выполнено в рамках инновационной образовательной программы Российского университета дружбы народов, направление «Комплекс экспортноориентированных инновационных образовательных программ по приоритетным направлениям науки и технологий», и входит в состав учебно-методического комплекса, включающего описание курса, программу и электронный учебник.

СОДЕРЖАНИЕ

1 НАЧАЛА MATHCAD	7
1.1 Назначение <i>Mathcad</i>	7
1.2 Создание документа на основе шаблона	9
1.3 Ввод и редактирование формул	11
1.3.1 Элементы интерфейса	12
1.3.2 Ввод формул	13
1.3.3 Изменение формул	14
1.3.4 Вырезка, копирование и вставка части формулы	16
1.3.5 Ввод символов, операторов и функций	16
1.3.6 Ввод и редактирование текста	18
1.3.7 Ввод текста	18
1.3.8 Редактирование текста	19
1.3.9 Импорт текста	20
1.3.10 Математические символы внутри текста	21
1.3.11 Гиперссылки	22
1.3.12 Посылка документа по электронной почте	22
1.4 ПЕРЕМЕННЫЕ И ФУНКЦИИ	22
1.4.1 Определение переменных	23
1.4.2 Присваивание переменным значений	23
1.4.3 Функции	24
1.4.4 Определение функции пользователя	25
1.4.5 Вывод значений переменных и функций	26
1.5 Операторы	27
1.5.1 Арифметические операторы	28
1.5.2 Вычислительные операторы	28

1.6	Матричные вычисления	30
1.6.1	Простейшие операции с матрицами	30
1.6.2	Транспонирование	31
1.6.3	Сложение	31
1.6.4	Векторизация массивов	32
1.7	Матричные функции	33
1.7.1	Функции создания матриц	34
1.7.2	Слияние и разбиение матриц	34
1.7.3	Вывод размера матриц	36
1.7.4	Сортировка матриц	37
1.8	Системы линейных алгебраических уравнений	38
2	ПРОГРАММИРОВАНИЕ	41
2.1	Программирование в <i>Mathcad</i>	41
2.2	Язык программирования <i>Mathcad</i>	42
2.2.1	Что такое программа?	43
2.2.2	Создание программы (<i>Add Line</i>)	44
2.2.3	Локальное присваивание	45
2.2.5	Условные операторы (<i>if, otherwise</i>)	45
2.2.6	Операторы цикла (<i>for, while, break, continue</i>)	46
2.2.7	Возврат значения (<i>return</i>)	48
2.2.8	Перехват ошибок (<i>on error</i>)	49
2.3	Примеры программирования	49
2.4	Интерполяция	51
2.4.1	Линейная интерполяция	51
2.4.2	Кубическая сплайн-интерполяция	53
3	ВВОД-ВЫВОД ДАННЫХ	56
3.1	Числовой ввод-вывод	56
3.2	Создание графиков	56

3.3	Двумерные графики	59
3.3.1	XУ-график двух векторов	59
3.3.2	XУ-график вектора и ранжированной переменной	60
3.3.3	XУ-график функции	60
3.3.4	Полярный график	62
3.3.5	Построение нескольких рядов данных	63
3.4	Ввод – вывод во внешние файлы	64
3.4.1	Текстовые файлы	65
3.4.2	Графические файлы	67
3.5	Ссылки и гиперссылки	67
3.5.1	Установка тега	68
3.5.2	Вставка гиперссылки	68
3.5.3	Ссылки	69
4	СИСТЕМЫ АВТОМАТИЗИРОВАННОГО РАСЧЕТА	71
4.1	Понятие систем автоматизированного расчета (САР)	71
4.2	Структура ППП (САР) и основные компоненты	76
4.3	Показатели качества ППП (САР)	79
4.4	Разработка и оформление модулей	84
5	ПОДГОТОВКА И ВВОД ИСХОДНЫХ ДАННЫХ	92
5.1	Программы <i>ID-HCDz</i>	92
5.2	Международная стандартная атмосфера	95
6	ПРОГРАММНЫЕ МОДУЛИ ТОПЛИВА И ТЕПЛОВОГО РАСЧЕТА	101
7	БЛОК ОКРУЖАЮЩЕЙ СРЕДЫ	112
7.1	Блок атмосферного давления	112
7.2	Блок начальной температуры	113

8 ПРОГРАММНЫЕ МОДУЛИ ДАВЛЕНИЯ НАДДУВА ДИЗЕЛЯ И КОЭФФИЦИЕНТА ИЗБЫТКА ВОЗДУХА	134
8.1 Блок давления наддува	134
8.2 Блок коэффициента избытка воздуха	134
9 ПРОГРАММНЫЕ МОДУЛИ ВЫСОТЫ ОПИСАНИЕ КУРСА И ПРОГРАММА	158 171

ТЕМА 1. НАЧАЛА *MATHCAD*

В данной главе рассмотрено назначение приложения *Mathcad* и, в целях знакомства с его основными возможностями, приведены базовые приемы его использования. В ней основное внимание уделено главным компонентам интерфейса *Mathcad*, который интуитивен и похож на другие программы *Windows* [1].

1.1. Назначение *Mathcad*

Mathcad является математическим редактором, позволяющим производить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов. Пользователи *Mathcad* — это студенты, ученые, инженеры, разнообразные технические специалисты. Благодаря простоте применения, наглядности математических действий, обширной библиотеке встроенных функций и численных методов, возможности символьных вычислений, а также превосходному аппарату представления результатов (графики самых разных типов, средства подготовки печатных документов и *Web*-страниц), стал наиболее популярным математическим приложением.

В отличие от большинства других современных математических приложений, *Mathcad* построен в соответствии с принципом: («Что Вы видите, то и получите»). Поэтому он очень прост в использовании, в частности, из-за отсутствия необходимости сначала писать программу, реализующую те или иные математические расчеты, а потом запускать ее на исполнение. Вместо этого достаточно просто вводить математические выражения с помощью встроенного редактора формул и тут же получать результат. Кроме того, можно изготовить на принтере печатную копию до-

кумента или создать страницу в Интернете именно в том виде, который этот документ имеет на экране компьютера при работе с *Mathcad*. Создатели *Mathcad* сделали все возможное, чтобы пользователь, не обладающий специальными знаниями в программировании, мог в полной мере приобщиться к достижениям современной вычислительной науки и компьютерных технологий.

В соответствии с проблемами реальной жизни, математикам приходится решать одну или несколько из следующих задач:

□ ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, *Web*-страниц);

- проведение математических расчетов;
- подготовка графиков с результатами расчетов;
- ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах;
- подготовка отчетов работы в виде печатных документов;
- подготовка *Web*-страниц и публикация результатов в Интернете;
- получение различной справочной информации из области математики.

Со всеми этими (а также некоторыми другими) задачами с успехом справляется *Mathcad*:

■ математические выражения и текст вводятся с помощью формульного редактора *Mathcad*, который по своим возможностям и простоте использования не уступает, к примеру, редактору формул, встроенному в *Microsoft Word*;

■ математические расчеты производятся немедленно, в соответствии с введенными формулами;

■ графики различных типов (по выбору пользователя) с возможностями форматирования вставляются непосредственно в документы;

- возможен ввод и вывод данных в файлы различных форматов;
- документы могут быть распечатаны непосредственно в *Mathcad* в том виде, который пользователь видит на экране компьютера, или сохранены в формате *RTF* для последующего редактирования в более мощных текстовых редакторах (например, *Microsoft Word*);
- возможно полноценное сохранение документов *Mathcad* в формате *Web*-страниц (генерация вспомогательных графических файлов происходит автоматически);
- имеется опция объединения разрабатываемых Вами документов в электронные книги, которые, с одной стороны, позволяют в удобном виде хранить математическую информацию, а с другой — являются полноценными *Mathcad*-программами, способными осуществлять расчеты;
- символьные вычисления позволяют осуществлять аналитические преобразования, а также мгновенно получать разнообразную справочную математическую информацию.

Таким образом, следует хорошо представлять себе, что в состав *Mathcad* входят несколько интегрированных между собой компонентов — это мощный текстовый редактор для ввода и редактирования как текста, так и формул, вычислительный процессор для проведения расчетов согласно введенным формулам, и символьный процессор, являющийся, по сути, системой искусственного интеллекта. Сочетание этих компонентов создает удобную вычислительную среду для разнообразных математических расчетов и, одновременно, документирования результатов работы.

1.2. Создание документа на основе шаблона

Для того чтобы создать новый пустой документ, уже работая в *Mathcad*, следует выполнить одно из трех эквивалентных действий:

- нажать одновременно клавиши $\langle Ctrl \rangle + \langle N \rangle$;

- нажать кнопку *New* (Создать) на панели инструментов;
- щелкнув на команде верхнего меню *File / New* (Файл / Создать). *Mathcad* позволяет создавать новые документы не с чистого листа, а на основе разработанных ранее. Для этого имеются два пути:
 - открыть ранее созданный документ и сохранить его под другим именем;
 - использовать шаблоны.

Шаблон (template) — это заготовка нового документа с введенными формулами, графиками, текстом, включая разметку, форматирование, выбор по умолчанию режима вычислений и т. д. В предыдущем разделе было рассмотрено создание нового документа на основе пустого шаблона. Пример документа, созданного на основе шаблона *\Web Page* (*\Web-страница*), показан на рис. 1.1. Новый документ, имеющий дизайн и характерную разметку *Web-страницы*, следует использовать, например, для представления расчетов в формате *HTML* в сети Интернет.

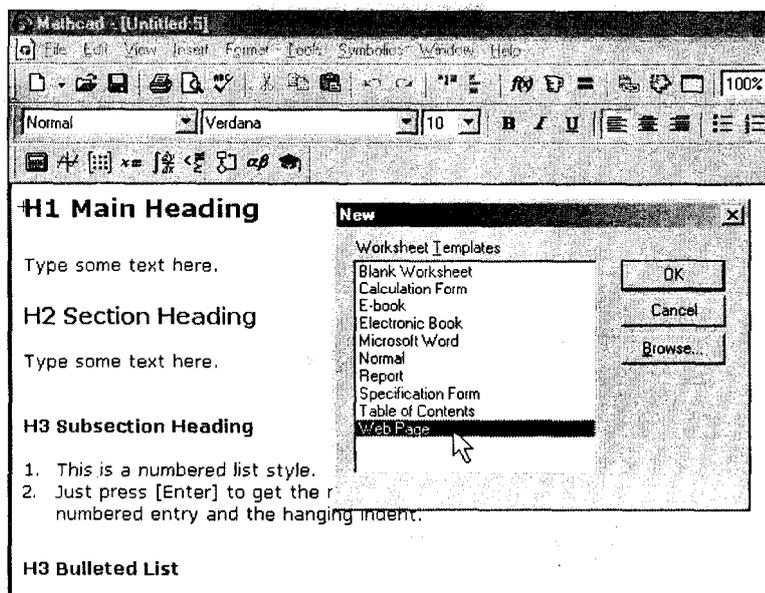


Рис. 1.1. Новый документ на основе шаблона *Web-страницы*

Для выбора другого шаблона, который имеется на компьютере в виде файла соответствующего формата, следует нажать в диалоговом ок-

не *NEW* (Создать) кнопку *Brows* (Обзор). В появившемся диалоге *Brows* (Обзор) найдите местоположение нужного файла с шаблоном *Mathcad*. Эти файлы имеют расширение *.mct* (*Math Cad Template*). Выберите желаемый шаблон в списке файлов и нажмите кнопку **Открыть** (*Open*).

В результате этих действий будет создан новый документ с уже имеющимися элементами оформления и определенными установками.

Для того чтобы разработать шаблон самостоятельно, можно проделать следующее:

1. Отредактировать документ, вводя формулы, текст, графики, отформатировать их и задать прочие установки документа.

2. Выполнить команду *File / Save As* (Файл / Сохранить как).

3. Выбрать в списке **Тип файла** (*File type*) элемент *Mathcad Template (*.mct)* (Шаблон *Mathcad*).

4. Ввести имя шаблона в поле ввода имени файла.

5. Нажать кнопку **Сохранить** (*Save*).

Впоследствии, созданный пользователем шаблон можно редактировать, открывая его как обычный документ *Mathcad*, помня лишь о том, что искать его надо среди файлов с расширением *.mct*.

1.3. Ввод и редактирование формул

Формульный редактор *Mathcad* позволяет быстро и эффективно вводить и изменять математические выражения. Тем не менее, некоторые аспекты его применения не совсем интуитивны, поэтому следует затратить определенное время на знакомство с особенностями формульного редактора, что впоследствии при реальной работе позволит сэкономить гораздо больше времени.

1.3.1. Элементы интерфейса

Перечислим элементы интерфейса *Mathcad* (рис. 1.2.)

- указатель мыши (*mouse pointer*) — играет обычную для приложений *Windows* роль, следуя за движениями мыши;

- курсор — обязательно находится внутри документа в одном из трех видов;

- курсор ввода (*crosshair*) — крестик красного цвета, который отмечает пустое место в документе, куда можно вводить текст или формулу;

- линии ввода (*editing lines*) — горизонтальная (*underline*) и вертикальная (*insertion line*) линии синего цвета, выделяющие в тексте или формуле определенную часть;

- линия ввода текста (*text insertion point*) — вертикальная линия, аналог линий ввода для текстовых областей.

- местозаполнители (*placeholders*) — появляются внутри незавершенных формул в местах, которые должны быть заполнены символом или оператором:

- местозаполнитель символа — черный прямоугольник;

- местозаполнитель оператора — черная прямоугольная рамка.

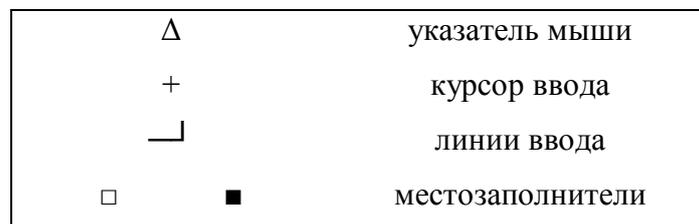


Рис. 1.2. Интерфейс редактирования

1.3.2. Ввод формул

Ввести математическое выражение можно в любом пустом месте документа *Mathcad*. Для этого следует поместить курсор ввода в желаемое место документа, щелкнув в нем мышью, и просто начинать вводить формулу, нажимая клавиши на клавиатуре. При этом в документе создается математическая область (*math region*), которая предназначена для хранения формул, интерпретируемых процессором *Mathcad*.

Покажем последовательность действий на примере ввода выражения

$\sin(x^{5+y})$:

+ $\sin(x$ $\sin(x^$ $\sin(x^{5+y}$ $\sin(x^{5+y})$

Рис. 1. 3. Пример ввода формулы

1. Щелкнуть мышью, обозначив место ввода.
2. Нажать последовательно клавиши $\langle s \rangle$, $\langle i \rangle$, $\langle n \rangle$, $\langle (\rangle \langle x \rangle$ — в этом месте вместо курсора ввода появится область (регион) с формулой, содержащей символы *sin(x*, причем она будет выделена линиями ввода.
3. Ввести оператор возведения в степень, нажав клавишу $\langle ^ \rangle$, либо выбрав кнопку возведения в степень на панели инструментов *Calculator* — в формуле появится местозаполнитель для введения значения степени, а линии ввода выделяют этот местозаполнитель.
4. Последовательно ввести остальные символы $\langle 5 \rangle$, $\langle + \rangle$, $\langle y \rangle$, $\langle) \rangle$.

Таким образом, поместить формулу в документе можно, просто начиная вводить символы, числа или операторы. Во всех этих случаях на месте курсора ввода создается математическая область с формулой, содержащей и линии ввода.

1.3.3. Изменение формул

Редактировать формулы в *Mathcad* можно так, как подсказывает интуиция и опыт работы с другими текстовыми редакторами. Большинство операций правки формул реализованы естественным образом. Однако некоторые из них несколько отличаются от общепринятых, что связано с особенностью *Mathcad* как вычислительной системы. Рассмотрим основные действия по изменению формул.

Вставка оператора

Операторы могут быть унарными (действующими на один операнд, как, например, оператор транспонирования матрицы или смены знака числа) и бинарными (например, + или /, действующими на два операнда). При вставке нового оператора в документ *Mathcad* определяет, сколько операндов ему требуется. Если в точке вставки оператор один или оба операнда отсутствуют, *Mathcad* автоматически помещает рядом с оператором один или два местозаполнители.

Следует иметь в виду, что то выражение в формуле, которое выделено линиями ввода в момент вставки оператора, становится его первым операндом.

Последовательность вставки оператора в формулу такова:

1. Поместить линии ввода на часть формулы, которая должна стать первым операндом.
2. Ввести оператор, нажав кнопку на панели инструментов или сочетание соответствующих клавиш.

Примечание. Для того чтобы вставить оператор перед частью формулы, выделенной линиями ввода, следует нажать перед его вводом клавишу *<Ins>*, которая передвинет вертикальную линию ввода вперед. Это важно, в частности, для вставки оператора отрицания.

Некоторые операторы *Mathcad* вставит в нужное место независимо

от положения линий ввода. Таков, например, оператор численного вывода \Rightarrow , который по смыслу выдает значение всей формулы в виде числа.

Выделение части формулы

Чтобы выделить часть формулы в некоторой математической области необходимо:

1. Поместить ее между линиями ввода, пользуясь, при необходимости, клавишами-стрелками и пробелом.

2. Поместить указатель мыши на вертикальную линию ввода, нажать и удерживать левую кнопку мыши.

3. Удерживая кнопку мыши, провести указатель мыши вдоль горизонтальной линии ввода, при этом часть формулы будет выделяться изменением цвета.

4. Отпустить кнопку мыши, когда будет выделена нужная часть формулы.

Примечание. Часть формулы можно выделить и без помощи мыши, нажимая клавиши со стрелками при удерживаемой клавише $\langle Shift \rangle$. В этом случае вместо перемещения линий ввода происходит выделение соответствующей части формулы. (Иногда работа с клавиатурой при выделении части математических областей более удобна.)

Удаление части формулы

Чтобы удалить часть формулы, следует:

1. Выделить ее (часть формулы).

2. Нажать клавишу $\langle Del \rangle$.

3. Кроме того, можно удалить часть формулы, помещая ее перед вертикальной линией ввода и нажимая клавишу $\langle BackSpace \rangle$.

Примечание. Имеется еще один способ удаления части формулы – выделить нужную часть, затем нажать комбинацию клавиш $\langle Ctrl \rangle + \langle X \rangle$, тем самым вырезая и помещая ее в буфер обмена. Этот способ удобен в случае, если требуется использовать фрагмент формулы в дальнейшем.

1.3.4. Вырезка, копирование и вставка части формулы

Для правки части формулы необходимо:

1. Выделить ее, либо просто поместить между линиями ввода, пользуясь мышью или клавишами-стрелками и пробелом.

2. Воспользоваться либо верхним меню *Edit* (Правка), либо контекстным меню, либо кнопкой на панели инструментов, или соответствующим сочетанием горячих клавиш:

- *Cut* (Вырезать) или $\langle \text{Ctrl} \rangle + \langle \text{X} \rangle$ — для вырезки части формулы в буфер;
- *Copy* (Копировать) или $\langle \text{Ctrl} \rangle + \langle \text{C} \rangle$ — для копирования в буфер;
- *Paste* (Вставить) или $\langle \text{Ctrl} \rangle + \langle \text{V} \rangle$ — для вставки из буфера предварительно помещенной туда части формулы.

Чтобы переместить (или скопировать) часть формулы из одной части документа в другую, вырежьте (скопируйте) ее в буфер обмена, перейдите к желаемому новому местоположению и вставьте ее туда из буфера.

1.3.5. Ввод символов, операторов и функций

Математические выражения содержат, как правило, самые различные, в том числе специфичные символы, набор которых в *Mathcad* выполняется не так, как в большинстве текстовых процессоров. Для вставки символов в документы доступны следующие инструменты:

■ большинство символов, например латинские буквы или цифры, для определения имен переменных и функций набираются на клавиатуре;

■ греческие буквы легче всего вставляются с помощью панели инструментов **Greek** (Греческие символы). Можно также ввести соответствующую латинскую букву и нажать клавиши <Ctrl>+<G> (после этого, например, из латинской буква «a» получается греческая «α»);

■ операторы могут быть вставлены либо с различных математических панелей инструментов, либо соответствующим сочетанием клавиш. Например, наиболее часто употребляемые операторы (см. рис.1.4.) сгруппированы на панели **Calculator** (Калькулятор);

Calculator					X
sin	cos	tan	ln	log	n!
i	x	√	ⁿ √	e ^x	$\frac{1}{x}$
()	x ²	x ^y	π	7	8
9	/	†	4	5	6
x	÷	1	2	3	+
:=	.	0	-	=	

Рис. 1.4. Панель **Calculator** :
шаблоны основных математических операций, цифр, знаков арифметических операций

■ имена функций вводятся либо с клавиатуры, либо, с большей надежностью, с помощью команды **Insert / Function** (Вставка / Функция);

■ скобки могут быть вставлены нажатием соответствующих клавиш. Однако для того чтобы выделить скобками уже введенную часть формулы, лучше поместить ее между линиями ввода и нажать клавишу <'> (апостроф).

1.3.6. Ввод и редактирование текста

Mathcad — это математический редактор. Основное его назначение заключается в редактировании математических формул и расчете по ним. Вместе с тем, наряду с формульным редактором *Mathcad* обладает довольно развитыми средствами по оформлению текста. Назначение текстовых областей в документах *Mathcad* для разных пользователей и разных задач может быть различным. Качественно стоит различать подход к тексту, используемому:

- просто в качестве комментариев;
- как элемент оформления документов для создания качественных отчетов в печатной и электронной формах.

Рассмотрим некоторые основные приемы работы с текстом (с расширенными возможностями текстового процессора *Mathcad* можно ознакомиться по [2, 3, 4 и др.]).

1.3.7. Ввод текста

Текстовую область (или, по-другому, регион с текстом — *text region*) можно разместить в любом незанятом месте документа *Mathcad*. Однако, когда пользователь помещает курсор ввода в пустое место документа и просто начинает вводить символы, *Mathcad* по умолчанию интерпретирует их как начало формулы. Чтобы до начала ввода указать программе, что требуется создать не формульный, а текстовый регион, достаточно, перед тем как ввести первый символ, нажать клавишу <">. В результате на месте курсора ввода появляется новый текстовый регион, который имеет характерное выделение (рис. 1.5). Курсор принимает при этом вид вертикальной линии красного цвета, которая называется *линией ввода текста* и аналогична по назначению линиям ввода в формулах.

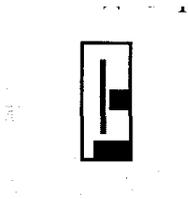


Рис. 1.5. Вновь созданная текстовая область
(*text region*)

Примечание. Создать текстовый регион можно и эквивалентным способом, с помощью команды *Insert / Text Region* (Вставка / Текстовая область).

Теперь можно просто вводить любой текст в текстовый регион.

1.3.8. Редактирование текста

Чтобы изменить какой-либо текст внутри документа, следует:

1. Щелкнуть мышью на области текста — она приобретет характерный вид (рис. 1.6.).

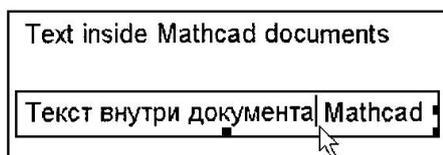


Рис. 1.6. Текстовые области (нижняя – в процессе редактирования)

2. При необходимости переместить линию ввода текста внутри текстовой области к символам, которые собираетесь изменить, щелкнув мышью в нужном месте текста или нажав клавиши со стрелками и клавиши <Home> и <End>.

3. Отредактировать текст.

Для редактирования текста применяются те же средства, что и для редактирования формул. Кроме того, существуют развитые средства фор-

матирования текста, такие, как управление типом и размером шрифта, выравниванием и т.п.

1.3.9. Импорт текста

Mathcad позволяет импортировать фрагменты текста из других приложений (например, *Notepad* или *Microsoft Word*). Сделать это проще всего через буфер обмена:

1. Находясь в другом приложении, можно скопировать нужный фрагмент в буфер обмена.

2. Перейдя в окно *Mathcad* и отметив (щелчком мыши в желаемом месте) курсором ввода место вставки фрагмента.

3. Можно выбрать один из двух путей:

- создать в документе *Mathcad* текстовый регион, нажав клавишу <">, и, находясь внутри него, вставить содержимое буфера обмена сочетанием <Ctrl>+<V>. Фрагмент будет вставлен в документ в виде обычной текстовой области, которую затем можно редактировать стандартными для *Mathcad* средствами;

- не создавая текстовой области, вставить фрагмент из буфера нажатием <Ctrl>+<V>. Текст будет вставлен в виде объекта *OLE* (Object Linking and Embedding), т.е. для его редактирования каждый раз будет вызываться то приложение, в котором он был создан.

Второй способ лучше применять в том случае, когда специальное форматирование текста невозможно осуществить средствами *Mathcad*. Чтобы вернуться к редактированию вставленного таким образом текста, следует дважды щелкнуть на нем мышью. При этом вместо меню и панелей инструментов в *Mathcad* появятся соответствующие средства другого приложения. С их помощью можно отредактировать объект и затем

вернуться к обычному редактированию документа *Mathcad*, щелкнув в нем мышью за пределами объекта.

1.3.10. Математические символы внутри текста

Для качественного оформления документов, вполне возможно, потребуются текстовые области, содержащие математические выражения. Для создания таких областей следует:

1. Щелкнуть в нужной части текстовой области.
2. Выбрать команду *Insert / Math Region* (Вставка / Математическая область) или нажмите клавиши $\langle \text{Ctrl} \rangle + \langle \text{Shift} \rangle + \langle \text{A} \rangle$, чтобы создать пустой местозаполнитель внутри текста (рис. 1.7.).
3. Ввести математическое выражение в местозаполнитель так, как вводятся обычные формулы.

Input

Рис. 1.7. Вставка математических символов в текстовую область

Помещая формулы в текст, следует помнить о том, что они влияют на вычисления точно так же, как если бы были помещены в математический регион непосредственно в документе. (После переопределения внутри текста переменная x меняет свое значение.)

Если необходимо, чтобы математическая область внутри текста не влияла на вычисления, можно отключить их. Для этого, находясь в режиме редактирования формулы, выполнить команду *Format / Properties* (Формат / Свойства) и, перейдя в открывшемся диалоговом окне *Properties* (Свойства) на вкладку *Calculations* (Вычисления), установите флажок *Disable Evaluations* (Выключить вычисления) и нажмите кнопку **ОК**.

1.3.11. Гиперссылки

Иногда необходимо сделать текстовую область одновременно и гиперссылкой, переводящей курсор на какое-либо иное место в активном документе, на другой документ *Mathcad*, либо на сайт в Интернете. Для вставки гиперссылки используется команда ***Insert / Hyperlink*** (Вставка / Гиперссылка).

1.3.12. Посылка документа по электронной почте

Послать документ по электронной почте можно с помощью почтового приложения (например, *Microsoft Outlook*), присоединяя файл с документом *Mathcad* к письму обычным образом. Но, кроме того, отправить активный документ по электронной почте легко, не выходя из *Mathcad*. Для этого следует выбрать команду ***File / Send*** (Файл / Отправить), в результате чего сразу появится окно ***NEW Message*** (Новое сообщение), с автоматически присоединенным к нему файлом *Mathcad*. Пользователю остается лишь ввести в соответствующие поля окна электронный адрес получателя, тему и текст письма (последние два пункта необязательны) и отправить письмо. (Для использования этой опции компьютер должен быть подключен к Интернету.)

1.4. ПЕРЕМЕННЫЕ И ФУНКЦИИ

Основные инструменты математика — это операции с переменными величинами и функциями. В *Mathcad* переменные, операторы и функции реализованы в интуитивной форме, т.е. выражения в редакторе вводятся и вычисляются так, как они были бы написаны на листе бумаги. Порядок вычислений в документе *Mathcad* также очевиден: математические выраже-

ния и действия воспринимаются процессором слева направо и сверху вниз.

Перечислим основные действия, которые пользователь может совершать для определения и вывода переменных и функций.

1.4.1. Определение переменных

Чтобы определить переменную, достаточно ввести ее имя и присвоить ей некоторое значение, для чего служит оператор присваивания.

1.4.2. Присваивание переменным значений

Чтобы присвоить переменной новое значение, например переменную y сделать равной **100**, необходимо:

1. Ввести в желаемом месте документа имя переменной, например y .

2. Ввести оператор присваивания с помощью клавиши $\langle := \rangle$ или нажатием соответствующей кнопки *Definition* (Присваивание) на панели инструментов *Calculator* или *Evaluation* (Выражение).

3. Введите в появившийся местозаполнитель значение переменной (**100**).

Результат поэтапных действий показан на рис. 1.8.

a		b		c
$y := \blacksquare$		$y := 100$		$s := \text{"Yes"}$
		$z := (y - 30)^3 + 10$		

Рис. 1.8. Присваивание переменным:
 a – числового значения; b – вычисляемого значения выражения; c – строкового значения

Ввести новое значение переменной возможно как в виде числа, так и в виде математического выражения, содержащего другие переменные и функции, а также в виде строкового выражения. В последнем случае будет создана переменная s не численного, а строкового типа (рис. 1.8).

Если переменная с некоторым именем создается в данном документе впервые, то для ввода оператора присваивания, вместо двоеточия, допускается использовать символ равенства “=”, который *Mathcad* автоматически заменит символом присваивания.

В некоторых случаях это невозможно, в частности, когда значение присваивается переменной, имя которой зарезервировано *Mathcad*. Например, присвоить значение переменной с именем N можно, лишь вводя двоеточие, т.к. по умолчанию это имя обозначает в *Mathcad* размерность силы (Ньютон).

Чтобы переопределить значение переменной, определенной в документе, оператор присваивания следует вводить не знаком равенства, а двоеточием, либо пользоваться панелью инструментов.

1.4.3. Функции

Функции в *Mathcad* записываются в обычной для математика форме, где:

- $f(x, \dots)$ - функция;
- f – имя функции;
- x, \dots – список переменных.

Легче всего ввести написание функции в документ при помощи клавиатуры.

В *Mathcad* формально можно разделить функции на два типа:

- встроенные;
- функции, определенные пользователем.

Применение функций обоих типов в расчетах совершенно одинаково, с тем исключением, что любую встроенную функцию можно сразу использовать в любом месте документа, а пользовательскую функцию необходимо предварительно определить в документе до момента вычисления ее значения.

1.4.4. Определение функции пользователя

Для определения функцию пользователя, например,

$$F(x,y) = x^2 \cdot \cos(x+y),$$

необходимо:

1. В желаемом месте документа ввести имя функции (f).
2. Затем левую скобку "(", имена переменных через запятую x , y и правую скобку ")". При вводе левой скобки и запятой автоматически будут появляться соответствующие местозаполнители.
3. Ввести оператор присваивания с панели инструментов или нажатием клавиши $\langle \rangle$.
4. В появившийся местозаполнитель ввести выражение, определяющее функцию $x^2 \cdot \cos(x+y)$, пользуясь клавиатурой или панелями инструментов.

Результат ввода показан на рис.1.9.

$$f(x,y):=x^2 \cdot \cos(x+y)$$

Рис. 1. 9. .Определение функции пользователя

Все переменные, присутствующие в выражении определения функции справа, либо должны входить в список аргументов функции (в скобках, слева после имени функции), либо должны быть определены ранее. В противном случае будет выведено сообщение об ошибке, причем имя неопределенной переменной будет выделено красным цветом.

1.4.5. Вывод значений переменных и функций

Чтобы вычислить в документе некоторое математическое выражение, которое может состоять из переменных, операторов и функций (встроенных и определенных пользователем), следует:

1. Ввести это выражение, например x^y .
2. Нажать клавишу $\langle = \rangle$.

В результате справа от введенного знака равенства появится вычисленное значение выражения (рис.1. 10.). Нельзя изменять содержимое выражения справа от знака равенства, поскольку оно есть результат работы вычислительного процессора *Mathcad*.

О том, что некоторое выражение документа находится в стадии вычисления, свидетельствует обрамляющая его зеленая рамка и невозможность предпринять какое-либо действие с программой *Mathcad*.

Перед тем, как вычислить значение математического выражения, необходимо определить значение каждой входящей в него переменной (две первых строки рис. 1.10а). Вычисляемое выражение может содержать практически любое количество переменных, операторов и функций. Возможные способы вывода текущего значения той или иной переменной приведены на рис. 1.10.

<p>a x:=10 y:=(x - 3)² + 1 x^y=1×10⁵⁰ x=10</p>	<p>b f(x,y):=x²·cos(x+y) f(2, 5.99)= -0.542 f(1.3,7)= -0.729</p>
<p>c x:=1.3 y:=7 f(x,y)= -0.729</p>	<p>d y:=5 f(x):= x²·cos(x+y) f(1)=0.96</p>
<p>e y:=0 x=1 f(x)=0.96 x²·cos(x+y)=0.54 x²·cos(x+5)=0.96</p>	

Рис. 1.10. Вывод значений:

a – вычисление выражения; *b* – вывод значения функции; *c* – вывод значения функции; *d* – определение функции пользователя; *e* – определение функции пользователя

1.5. Операторы

Каждый оператор в *Mathcad* обозначает некоторое математическое действие в виде символа. В полном согласии с терминологией, принятой в математике, ряд действий (например, сложение, деление, транспонирование матрицы и т. п.) реализован в *Mathcad* в виде встроенных операторов, а другие действия (например, *sin*, *erf* и т. п.) — в виде встроенных функций. Каждый оператор действует на одно или два числа (переменную или функцию), которые называют *операндами*. Если в момент вставки оператора одного или обоих операндов не хватает, то недостающие операнды будут отображены в виде местозаполнителей. Символ любого оператора в нужное место документа вводится одним из двух основных способов:

- нажатием соответствующей клавиши (или сочетания клавиш) на клавиатуре;

- нажатием указателем мыши соответствующей кнопки на одной из математических панелей инструментов.

Напомним, что большинство математических панелей содержат сгруппированные по смыслу математические операторы, а вызвать эти панели на экран можно нажатием соответствующей кнопки на панели *Math* (Математика).

1.5.1. Арифметические операторы

Операторы, обозначающие основные арифметические действия, вводятся с панели *Calculator* (Калькулятор):

- сложение и вычитание: $+$ $-$;
- умножение и деление: \cdot $/$;
- факториал: $!$; $.$
- модуль числа: $|x|$;
- квадратный корень: \sqrt{x} ;
- корень n -й степени: $\sqrt[n]{}$;
- возведение x в степень y : x^y ;
- изменение приоритета: скобки;
- численный вывод: $=$.

1.5.2. Вычислительные операторы

Вычислительные операторы вставляются в документы при помощи панели инструментов *Calculus* (Вычисления). При нажатии любой из кнопок в документе появляется символ соответствующего математического

действия, снабженный несколькими местозаполнителями. Количество и расположение местозаполнителей определяются типом оператора и в точности соответствуют их общепринятой математической записи. Например, при вставке оператора суммы (рис. 1.11) необходимо задать четыре величины: переменную, по которой надо произвести суммирование, нижний и верхний пределы, а также само выражение, которое будет стоять под знаком суммы (пример заполненного оператора суммы см. на рис.1.11).

Для того чтобы вычислить неопределенный интеграл, следует заполнить два местозаполнителя: подынтегрального выражения и переменной интегрирования.

После ввода какого-либо вычислительного оператора имеется возможность вычислить его значение либо численно, нажатием клавиши $\langle \Rightarrow \rangle$, либо символично, с помощью оператора символического вывода.

a Операторы вычисления производных

$$\frac{d}{dx} \cos(x) \rightarrow -\sin(x)$$

$$\frac{d^2}{dx^2} \sin(x) \rightarrow -\sin(x)$$

b Операторы интегрирования

$$\int_a^{\infty} \frac{1}{x^3} dx \rightarrow \frac{1}{2 \cdot a^2}$$

$$\int \ln(x) dx \rightarrow x \cdot \ln(x) -$$

c Операторы суммирования и вычисления произведения

$$\sum_{i=1}^{10} i = 55$$

$$\sum_{i=1}^{10} i \rightarrow 55$$

$$\prod_{i=1}^{30} i = 2.653 \times 10^{32}$$

d Операторы суммирования и произведения ранжированной переменной

$$i := 1..5$$

$$\sum_i i^2 \cdot i := 3.447 \times 10^3$$

$$\prod_i e^i = 3.269 \times 10^6$$

Рис. 1.11. Некоторые вычислительные операторы и примеры их применения

1.6. Матричные вычисления

Матричные вычисления можно условно разделить на несколько типов.

Первый — это простейшие действия, которые реализованы операторами и несколькими функциями, предназначенными для создания, объединения, сортировки, получения основных свойств матриц и т. п.

Второй — это более сложные функции, которые реализуют алгоритмы вычислительной линейной алгебры, такие, как решение систем линейных уравнений, вычисление собственных векторов и собственных значений, различные матричные разложения

1.6.1. Простейшие операции с матрицами

Простейшие операции матричной алгебры реализованы в *Mathcad* в виде операторов. Написание операторов по смыслу максимально приближено к их математическому действию. Каждый оператор выражается соответствующим символом.

Матричные и векторные операции *Mathcad*. Векторы являются частным случаем матриц размерности $n \times 1$, поэтому для них справедливы все те операции, что и для матриц, если ограничения особо не оговорены (например некоторые операции применимы только к квадратным матрицам $n \times n$). Какие-то действия допустимы только для векторов (например, скалярное произведение), а какие-то, несмотря на одинаковое написание, по-разному действуют на векторы и матрицы.

Непосредственное проведение векторных операций над строками, т. е. матрицами $1 \times n$, невозможно. Для того чтобы превратить строку в вектор, строку следует предварительно транспонировать.

1.6.2. Транспонирование

Транспонированием называют операцию, переводящую матрицу размерности $m \times n$ в матрицу размерности $n \times m$, делая столбцы исходной матрицы строками, а строки — столбцами. Ввод символа транспонирования (*transpose*) осуществляется с помощью панели инструментов *Matrix* (Матрица) или нажатием клавиш $\langle \text{Ctrl} \rangle + \langle 1 \rangle$.

1.6.3. Сложение

В *Mathcad* можно как складывать матрицы, так и вычитать их друг из друга.

a Транспонирование векторов и матриц

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}^T = (1 \quad 2 \quad 3)$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

b Сложение и вычитание матриц

$$\mathbf{A} := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$\mathbf{B} := \begin{pmatrix} 1 & 0 & 0 \\ -1 & -3 & -4 \end{pmatrix}$$

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} 2 & 2 & 3 \\ 3 & 2 & 2 \end{pmatrix}$$

$$\mathbf{A} - \mathbf{B} = \begin{pmatrix} 0 & 2 & 3 \\ 5 & 8 & 10 \end{pmatrix}$$

c Сложение матрицы со скаляром

$$\mathbf{A} := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad \mathbf{x} := 1$$

$$\mathbf{A} + \mathbf{x} = \begin{pmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix}$$

$$\mathbf{A} - \mathbf{x} = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$

d Умножение на скаляр

$$\mathbf{A} \cdot 2 = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$$

$$\frac{\mathbf{A}}{2} = \begin{pmatrix} 0.5 & 1 & 1.5 \\ 2 & 2.5 & 3 \end{pmatrix}$$

Рис. 1.12. некоторые простейшие операции с матрицами

Для этих операторов применяются символы $\langle + \rangle$ или $\langle - \rangle$ соответственно. Матрицы должны иметь одинаковую размерность.

Кроме сложения матриц, *Mathcad* поддерживает операцию сложения матрицы со скаляром (рис. 1.12d). Каждый элемент результирующей матрицы равен сумме соответствующего элемента исходной матрицы и скалярной величины.

Для того чтобы изменить знак матрицы, достаточно ввести перед ней знак минуса, как перед обычным числом.

О других возможных операциях над матрицами см. стр. 217–222 [1].

1.6.4. Векторизация массивов

Векторная алгебра *Mathcad* включает несколько необычный оператор, который называется *оператором векторизации* (*vectorize operator*). Этот оператор предназначен, как правило, для работы с массивами. Он позволяет провести однотипную операцию над всеми элементами массива (т.е. матрицы или вектора), упрощая тем самым программирование циклов. Например, требуется умножить каждый элемент одного вектора на соответствующий элемент другого вектора. Непосредственно такой операции в *Mathcad* нет, но ее легко осуществить с помощью векторизации (рис. 1.13). Для этого следует:

1. Ввести векторное выражение.
2. Переместить курсор таким образом, чтобы линии ввода выделяли все выражение, которое требуется подвергнуть векторизации.
3. Ввести оператор векторизации кнопкой **Vectorize** (Векторизация) на панели **Matrix** (Матрица), или сочетанием клавиш $\langle \text{Ctrl} \rangle + \langle - \rangle$.
4. Ввести $\langle = \rangle$, чтобы получить результат.

Оператор векторизации можно использовать только с векторами и матрицами одинакового размера.

Некоторые неспецифические функции *Mathcad* не требуют векторизации для проведения одной и той же операции над всеми элементами вектора. Например, аргументом тригонометрических функций по определению является скаляр. Если попытаться вычислить синус векторной величины, *Mathcad* осуществит векторизацию по умолчанию, вычислив синус каждого элемента и выдав в качестве результата соответствующий вектор.

$$v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad v \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 14 \quad \left(\overrightarrow{\left[v \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \right]} \right) = \begin{pmatrix} 1 \\ 4 \\ 9 \end{pmatrix}$$

Рис. 1.13. Использование векторизации для перемножения элементов вектора

$$\sin(v) = \begin{pmatrix} 0.841 \\ 0.909 \\ 0.141 \end{pmatrix} \quad \sin(\overrightarrow{v}) = \begin{pmatrix} 0.841 \\ 0.909 \\ 0.141 \end{pmatrix}$$

Рис. 1.14. Векторизация необязательна для некоторых функций *Mathcad*

Все матричные и векторные операторы могут быть использованы в символьных вычислениях (см. стр. 224–225, [1]).

1.7. Матричные функции

Перечислим основные встроенные функции, предназначенные для облегчения работы с векторами и матрицами. Они нужны для создания матриц, слияния и выделения части матриц, получения основных свойств матриц и т. п.

1.7.1. Функции создания матриц

Самым наглядным способом создания матрицы или вектора является применение первой кнопки панели инструментов *Matrix* (Матрицы). Однако в большинстве случаев, в частности, при программировании сложных проектов, удобнее бывает создавать массивы с помощью встроенных функций.

Определение элементов матрицы через функцию

■ `matrix(M,N,f)` — создание матрицы размера $M \times N$, каждый i,j элемент которой есть $f(i,j)$ (рис. 1.15):

- M — количество строк;
- N — количество столбцов;
- $f(i,j)$ — функция.

$$f(i,j) := i + 0.5 \cdot j \quad A := \text{matrix}(2,3,f)$$

$$A = \begin{pmatrix} 0 & 0.5 & 1 \\ 1 & 1.5 & 2 \end{pmatrix}$$

Рис. 1.15. Создание матрицы

Для создания матриц имеются еще две специфические функции, применяемые, в основном, для быстрого и эффектного представления каких-либо зависимостей в виде трехмерных графиков (типа поверхности или пространственной кривой) (стр.226–227 [1]).

1.7.2. Слияние и разбиение матриц

Из матриц или вектора можно выделить: подматрицу, вектор-столбец, элемент, а также объединить несколько матриц в одну.

Выделение части матрицы

Часть матрицы выделяется одним из следующих способов:

■ для выделения одного элемента предназначен оператор нижнего индекса. Оператор вводится нажатием кнопки *Subscript* (Нижний индекс) со значком x_n на панели *Matrix* (Матрица), либо нажатием клавиши $\langle \rangle$;

■ для выделения из матрицы столбца используется оператор выделения столбца нажатием кнопки *Matrix Column* с изображением угловых скобок $\langle \rangle$ на панели *Matrix*, либо сочетанием клавиш $\langle \text{Ctrl} \rangle + \langle 6 \rangle$ (рис. 1.16). Этот оператор называют еще, по аналогии с предыдущим, оператором верхнего индекса;

■ чтобы выделить из матрицы строку, можно применить тот же оператор $\langle \rangle$ к транспонированной матрице (рис. 1.17);

■ для выделения подматрицы используется встроенная функция $submatrix(A, ir, jr, ic, jc)$, которая возвращает часть матрицы A , находящуюся между строками ir, jr и столбцами ic, jc включительно (рис. 1.17).

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$
$$A_{0,1} = 2 \quad A_{1,1} = 5$$
$$A^{\langle 2 \rangle} = \begin{pmatrix} 3 \\ 6 \end{pmatrix} \quad A^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 4 \end{pmatrix} \quad A^{\langle 1 \rangle} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

Рис. 1. 16. Доступ к отдельным элементам, столбцам и строкам матрицы

Выделить из матрицы один столбец или строку можно и с помощью функции *submatrix*.

Те же операции применимы к матрицам-векторам и матрицам-строкам. Следует помнить только, что размер их составляет $N \times 1$ и $1 \times N$, соответственно (рис. 1.18).

Слияние матриц

Для того, чтобы составить из двух или более матриц одну, в *Mathcad* предусмотрены две матричные функции (рис. 1.19):

■ *Augment* (A, B, C, \dots) – матрица, сформированная слиянием матриц – аргументов слева направо;

■ *stack*(A, B, C, \dots) – матрица, сформированная слиянием матриц – аргументов сверху вниз;

A, B, C, \dots – векторы или матрицы соответствующего размера.

$$\mathit{submatrix}\left[\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, 0, 1, 0, 1\right] = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$$

$$\mathit{submatrix}\left[\begin{pmatrix} 1 & 0 & 0 \\ -1 & -3 & -4 \end{pmatrix}, 0, 0, 0, 1\right] = (1 \ 0)$$

Рис. 1.17. Выделение подматриц

$$(1 \ 2 \ 3) \langle 0 \rangle = (1) \quad \left(\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \right)^T \langle 1 \rangle = (2)$$

$$\mathit{submatrix}\left[\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, 0, 1, 0, 0\right] = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Рис. 1.18. Выделение частей из векторов и строк

1.7.3. Вывод размера матриц

Для получения сведений о характеристиках матриц или векторов предусмотрены следующие встроенные функции (рис. 1.20.):

■ *rows*(A) – число строк;

- $cols(A)$ – число столбцов;
 - $length(v)$ – число элементов вектора;
 - $last(v)$ – индекс последнего элемента вектора.
- A – матрица или вектор;
 - v – вектор.

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad B := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$stack(A, B) = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad augment(A, B) = \begin{pmatrix} 1 & 2 & 3 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 \end{pmatrix}$$

Рис. 1.19. Примеры слияния матриц

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \quad v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad w := (1 \ 2 \ 3)$$

$$rows(A) = 3 \quad rows(v) = 3 \quad rows(w) = 1$$

$$cols(A) = 2 \quad cols(v) = 1 \quad cols(w) = 3$$

$$length(v) = 3 \quad last(v) = 2$$

Рис. 1.20. Размер матриц и векторов

1.7.4. Сортировка матриц

В процессе расчетов иногда необходимо переставить элементы матрицы или вектора, расположив их в определенной строке или столбце в порядке возрастания или убывания. Для этого имеются несколько встроенных функций, которые позволяют оперативно управлять работой с матрицами:

■ $sort(v)$ – сортировка элементов вектора в порядке возрастания (рис. 1.21).

• v – вектор.

■ $csort(A,i)$ – сортировка строк матрицы выстраиванием элементов i -го столбца в порядке возрастания.

• v – вектор;

• A – матрица;

■ $rsort(A,i)$ – сортировка столбцов матрицы выстраиванием элементов i -ой строки в порядке возрастания;

■ $reverse(v)$ – перестановка элементов вектора в обратном порядке.

$$v := \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix} \quad sort(v) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad reverse(v) = \begin{pmatrix} 2 \\ 4 \\ 1 \\ 3 \end{pmatrix}$$

$$A := \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 8 \end{pmatrix} \quad csort(A,1) = \begin{pmatrix} 3 & 0 \\ 2 & 8 \\ 1 & 9 \end{pmatrix} \quad csort(A,0) = \begin{pmatrix} 1 & 9 \\ 2 & 8 \\ 3 & 0 \end{pmatrix}$$

$$rsort(A,1) = \begin{pmatrix} 9 & 1 \\ 0 & 3 \\ 8 & 2 \end{pmatrix} \quad rsort(A,2) = \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 8 \end{pmatrix}$$

Рис. 1.21. Сортировка векторов, матриц по столбцу, а также матриц по строке

1.8. Системы линейных алгебраических уравнений

Центральным вопросом вычислительной линейной алгебры является решение систем линейных алгебраических уравнений (СЛАУ), т. е. систем уравнений вида

$$a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \dots + a_{in} \cdot x_n = b_i \quad (1)$$

В матричной форме СЛАУ записываются в эквивалентном виде:

$$A \cdot x = b, \quad (2)$$

где A — матрица коэффициентов СЛАУ размерности $M \times N$; x — вектор неизвестных; b — вектор правых частей уравнений.

СЛАУ имеют единственное решение, если матрица A является невырожденной, или, по-другому, несингулярной, т.е. ее определитель не равен нулю. С вычислительной точки зрения, решение СЛАУ не представляет трудностей, если матрица A не очень велика. С большой матрицей проблем также не возникнет, если она достаточно обусловлена. В *Mathcad* СЛАУ можно решить как в более наглядной (1), так и в более удобной для записи форме (2). Для первого способа следует использовать вычислительный блок (рис. 1.22), а для второго — встроенную функцию:

■ $lsolve(A, b)$ — решение системы линейных уравнений;

- A — матрица коэффициентов системы;
- b — вектор правых частей.

Применение функции $lsolve$ показано на рис. 1.22.

В некоторых случаях, для большей наглядности представления СЛАУ, его можно решить точно так же, как систему нелинейных уравнений (рис. 1.23.).

$$A := \begin{pmatrix} 1 & 5 & 2 \\ 0.7 & 12 & 5 \\ 3 & 0 & 4 \end{pmatrix} \quad b := \begin{pmatrix} 1 \\ 2.9 \\ 3.1 \end{pmatrix}$$

$$lsolve(A, b) = \begin{pmatrix} -0.186 \\ -0.129 \\ 0.915 \end{pmatrix}$$

Рис. 1.22. Решение системы линейных алгебраических уравнений

Решение СЛАУ с помощью вычислительного блока:

$$x := 0 \quad y := 0 \quad z := 0$$

Given

$$1 \cdot x + 5 \cdot y + 2 \cdot z = 1$$

$$0.7 \cdot x + 12 \cdot y + 5 \cdot z = 2.9$$

$$3 \cdot x + 0 \cdot y + 4 \cdot z = 3.1$$

$$\mathit{find}(x, y, z) = \begin{pmatrix} -0.186 \\ -0.129 \\ 0.915 \end{pmatrix}$$

Рис. 1.23. Решение СЛАУ с помощью вычислительного блока

ТЕМА 2. ПРОГРАММИРОВАНИЕ

Разработчики *Mathcad* изначально поставили перед собой такую задачу, чтобы дать возможность профессионалам-математикам, физикам и инженерам самостоятельно проводить сложные расчеты, не обращая за помощью к программистам. Опыт показал, что без программирования *Mathcad* серьезно теряет в своей силе, в основном, из-за недовольства пользователей, знакомых с техникой создания программ и желающих осуществить свои расчеты в привычном для себя программистском стиле. Вместо знакомых принципов программирования, пользователям старых версий *Mathcad* предлагалось комбинировать несколько специфических встроенных функций и ранжированные переменные.

Последние версии *Mathcad* имеют не очень мощный, но весьма элегантный собственный язык. С одной стороны, он дает возможность программисту эффективно применять программный код в документах *Mathcad*. С другой – простота и интуитивность языка программирования позволяет быстро ему обучиться. Наконец, программные модули внутри документа *Mathcad* сочетают в себе и обособленность (поэтому их легко отличить от остальных формул), и простоту смыслового восприятия.

Несмотря на небольшое число операторов, язык программирования *Mathcad* позволяет решать самые различные, в том числе и довольно сложные, задачи.

2.1. Программирование в *Mathcad*

В ранних версиях *Mathcad* встроенного языка программирования не было. Чтобы применять привычные операции проверки ус-

ловий и организовывать циклы, приходилось применять сложные алгоритмы из встроенных функций условия *if* и *until* и комбинаций ранжированных переменных (рис. 2.1).

$$\begin{aligned} f(x) &:= \text{if}(x < 0, "negative", "positive") \\ f(1) &= "positive" \\ f(-1) &= "negative" \end{aligned}$$
$$i := 0..10$$
$$x_i := i^2$$

Рис. 2.1. Функция условия *if*; организация цикла при помощи ранжированной переменной x_i

Фактически, использование ранжированных переменных — мощный аппарат *Mathcad*, похожий на применение циклов в программировании. В подавляющем большинстве случаев намного удобнее организовать циклы (в том числе вложенные) с помощью ранжированных переменных, чем заниматься для этого программированием. Полезнее освоить технику, связанную с ранжированными переменными, векторами и матрицами, поскольку на ней основаны главные принципы расчетов в *Mathcad*, в частности подготовка графиков.

2.2. Язык программирования *Mathcad*

Для вставки программного кода в документы в *Mathcad* имеется специальная панель инструментов ***Programming*** (Программирование), которую можно вызвать на экран нажатием кнопки ***Programming Toolbar*** на панели ***Math*** (Математика).

Рассмотрим основные составные элементы языка программирования *Mathcad*.

2.2.1. Что такое программа?

Основными инструментами работы в *Mathcad* являются математические выражения, переменные и функции. Нередко записать формулу, использующую ту или иную внутреннюю логику (например, возвращение различных значений в зависимости от условий), в одну строку не удастся. Назначение программных модулей как раз и заключается в определении выражений, переменных и функций в несколько строк, часто с применением специфических программных операторов (рис. 2.2).

$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{"positive"} & \text{if } x > 0 \\ \text{"zero"} & \text{otherwise} \end{cases}$$
$$f(1) = \text{"positive"}$$
$$f(-1) = \text{"negative"}$$
$$f(0) = \text{"zero"}$$

Рис. 2.2. Функция условия, определенная с помощью программы

Несмотря на принципиальную эквивалентность определения функций и переменных через встроенные функции *Mathcad* или программные модули, программирование имеет ряд существенных преимуществ, которые в ряде случаев делают документ более простым и читаемым:

- возможность применения циклов и условных операторов;
- простота создания функций и переменных, требующих нескольких простых шагов (рис. 2.3);
- возможность создания функций, содержащих закрытый для остального документа код, включая преимущества использования локальных переменных и обработку исключительных ситуаций (ошибок).

Как видно из рис. 2.3, программный модуль обозначается в *Mathcad* вертикальной чертой, справа от которой последовательно записываются операторы языка программирования.

2.2.2. Создание программы (*Add Line*)

Чтобы создать программный модуль, следует:

1. Ввести часть выражения, которая будет находиться слева от знака присваивания, и сам знак присваивания. В примере это имя функции $f(x)$.

2. При необходимости можно вызвать на экран панель инструментов *Programming* (Программирование) и нажать на этой панели кнопку *Add Line* (Добавить линию).

3. Если приблизительно известно, сколько строк будет содержать программа, можно создать нужное количество линий повторным нажатием кнопки *Add Line* соответствующее число раз (на рис. 2.4 показан результат трехкратного нажатия).

4. В появившиеся местозаполнители ввести желаемый программный код, используя программные операторы. В примере в каждый местозаполнитель вводится строка, например, "*positive*" (рис. 2.3), затем нажимается кнопка *if* (Если) на панели *Programming* (Программирование) и в возникший местозаполнитель вводится выражение $x > 0$.

$$f(x) := \begin{array}{|l} \bullet \\ \bullet \\ \bullet \end{array} \qquad f(x) := \begin{array}{|l} \text{"negative" if } x > 0 \\ \text{"positive" if } \bullet \\ \text{"zero"} \end{array}$$

Рис. 2. 3. Начало создания программного модуля и вставка программного модуля

После того, как программный модуль полностью определен, и ни один местозаполнитель не остался незаполненным, функция может использоваться обычным образом как в численных, так и в символьных расчетах.

2.2.3. Локальное присваивание

Язык программирования *Mathcad* позволяет создавать внутри программных модулей локальные переменные, которые не воспринимаются из других частей документа. Присваивание в пределах программ, в отличие от документов *Mathcad*, производится с помощью оператора ***Local Definition*** (Локальное присваивание), который вставляется нажатием кнопки с изображением стрелки \leftarrow на панели ***Programming*** (Программирование).

Локальное присваивание показано на рис. 2.4.

$$f(x) := \left| \begin{array}{l} z \leftarrow 4 \\ z + x \end{array} \right. \quad f(1) = 5$$

Рис. 2.4. Локальное присваивание в программе

Переменная z существует только внутри программы, выделенной вертикальной чертой. Из других мест документа получить ее значение невозможно.

2.2.5. Условные операторы (*if, otherwise*)

Действие условного оператора *if* состоит из двух частей.

Сначала проверяется логическое выражение (условие) справа от него. Если оно истинно, выполняется выражение слева от оператора *if*. Если ложно — ничего не происходит, а выполнение программы продолжается переходом к ее следующей строке.

Вставить условный оператор в программу можно следующим образом (рис. 2.5):

1. Если необходимо, ввести левую часть выражения и оператор присваивания.

2. Создать новую строку программного кода, нажав на панели **Programming** (Программирование) кнопку **Add Line** (Добавить строку).

3. Нажать кнопку условного оператора *if*.

4. Справа от оператора *if* введите условие. Следует использовать логические операторы, вводя их с панели **Boolean** (Булевы операторы).

5. Выражение, которое должно выполняться, если условие истинно, ввести слева от оператора *if*.

6. Если в программе предусматриваются дополнительные условия, следует добавить в программу еще одну строку нажатием кнопки **Add Line** и ввести их таким же образом, используя оператор *if* или *otherwise*.

$$f(x) := \begin{cases} 0 & \text{if } x = 0 \\ \bullet & \text{if } \bullet \end{cases}$$

Рис. 2.5. Вставка условного оператора

Оператор *otherwise* используется совместно с одним или несколькими условными операторам *if* и указывает на выражение, которое будет выполняться, если ни одно из условий не оказалось истинным. Примеры использования операторов *if* и *otherwise* приведены на рис. 2.2.

2.2.6. Операторы цикла (*for, while, break, continue*)

В языке программирования *Mathcad* имеются два оператора цикла: *for* и *while*.

Первый из них дает возможность организовать цикл по некоторой переменной, заставляя ее принимать ряд последовательных значений.

Второй – создает цикл с выходом из него по некоторому логическому условию.

Чтобы вставить в программный модуль оператор цикла следует:

1. Создать в программном модуле новую линию.

2. Вставить один из операторов цикла *for* или *while* нажатием одноименной кнопки на панели **Programming** (Программирование).

3. Если выбран оператор *for* (рис. 2.6), то нужно вставить в соответствующие местозаполнители имя переменной и диапазон ее значений (рис. 2.7), а если *while* — то логическое выражение, при нарушении которого должен осуществляться выход из цикла (рис. 2.7).

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \mathit{for} \quad \bullet \in \bullet \\ \bullet \end{array} \right.$$

Рис. 2.6. Вставка оператора цикла

4. В нижний местозаполнитель ввести тело цикла, т.е. выражения, которые должны выполняться циклически.

При необходимости дополнить программу другими строками и ввести в них нужный код.

Иногда необходимо досрочно завершить цикл, т.е. не по условию в его заголовке, а в некоторой строке в теле цикла. Для этого предназначен оператор *break*.

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \mathit{for} \quad i \in 0..5 \\ \quad z \leftarrow z + 1 \end{array} \right.$$

$x = 15$

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \mathit{for} \quad i \in (1 \ 2 \ 3) \\ \quad z \leftarrow z + 1 \end{array} \right.$$

$x = 6$

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \mathit{while} \quad z < 10 \\ \quad z \leftarrow z + 1 \end{array} \right.$$

$x = 10$

Рис. 2.7. Оператор цикла *for* с ранжированной переменной. Оператор цикла *for* с вектором. Оператор цикла *while*

2.2.7. Возврат значения (*return*)

Если для определения переменной или функции применяется программный модуль, то его строки исполняются последовательно при вычислении в документе этой переменной или функции. Соответственно, по мере выполнения программы рассчитываемый результат претерпевает изменения. В качестве окончательного результата выдается последнее присвоенное значение. Чтобы подчеркнуть возврат программным модулем определенного значения, можно взять за правило делать это в последней строке программного модуля (рис. 2.8).

$$f(x) := \begin{array}{|l} y \leftarrow x^2 \\ z \leftarrow y + 1 \\ z \end{array}$$
$$f(2) = 5$$

Рис. 2.8. Возврат значения определен явно в последней строке программы

Вместе с тем, можно прервать выполнение программы в любой ее точке (например, с помощью условного оператора) и выдать некоторое значение, применив оператор *return*. В этом случае при выполнении указанного условия (рис. 2.9) значение, введенное в местозаполнитель после *return*, возвращается в качестве результата, и никакой другой код больше не выполняется. Вставляется в программу оператор *return* с помощью одноименной кнопки панели **Programming**.

$$f(x) := \left\{ \begin{array}{l} z \leftarrow x^2 \\ \text{return "zero" if } x = 0 \\ \text{return "i" if } x = i \\ z \end{array} \right.$$

$$\begin{aligned} f(-1) &= 1 \\ f(2) &= 4 \\ f(0) &= \text{"zero"} \\ f(i) &= \text{"i"} \end{aligned}$$

Рис. 2.9. Применение оператора *return*

2.2.8. Перехват ошибок (*on error*)

Программирование в *Mathcad* позволяет осуществлять дополнительную обработку ошибок. Если пользователь предполагает, что выполнение кода в каком-либо месте программного модуля способно вызвать ошибку (например деление на ноль), то эту ошибку можно перехватить с помощью оператора *on error*. Чтобы вставить его в программу, надо поместить линии ввода в ней в нужное положение и нажать кнопку с именем оператора *on error* на панели **Programming**. В результате появится строка с двумя местозаполнителями и оператором *on error* посередине (рис. 2.10).

2.3. Примеры программирования

Рассмотрим два простых примера использования программных модулей в *Mathcad* для численных (рис. 2.11.) и символьных (рис. 2.12.) расчетов. Следует помнить, что когда разрабатываются программные модули в *Mathcad*, операторы программирования вставляются в текст программы с помощью кнопок панели инструментов **Programming**. Их имена нельзя ни в коем случае просто набивать на клавиатуре, поскольку они не будут восприняты *Mathcad* корректно.

$$f(n) := \left| \begin{array}{l} z \leftarrow n \\ \text{"user_error : can't_divide_by_zero"} \quad \text{on error} \quad \frac{1}{z} \end{array} \right.$$

$$f(-2) \rightarrow \frac{-1}{2}$$

$$f(0) = \text{"user_error : can't_divide_by_zero"}$$

Рис. 2.10. Перехват ошибки деления на нуль

С помощью средств программирования можно создавать намного более сложные программы. Несколько примеров достаточно эффективного применения программирования даны в соответствующей литературе.

$$f(n) := \left| \begin{array}{l} \text{return } -99 \quad \text{if } n < 0 \\ z \leftarrow 1 \\ \text{for } i \in 1..n \\ \quad z \leftarrow z \cdot i \\ z \end{array} \right.$$

$$f(-2) \rightarrow -99$$

$$f(0) = 0$$

$$f(3.9) = 6$$

$$f(3) = 6$$

$$f(10) = 3.629 \times 10^6$$

Рис. 2.11. Программирование в численных расчетах

$$f(n) := \begin{cases} -1 & \text{if } n < 0 \\ x & \text{on error} \\ \frac{d^n}{dx^n} x^{10} & \text{otherwise} \end{cases}$$

$$f(1) \rightarrow 10 \cdot x^9$$

$$f(10) \rightarrow 3628800$$

$$f(-3) \rightarrow -1$$

$$f(2.1) \rightarrow x$$

Рис. 2.12. Программирование в символьных расчетах

2.1. Интерполяция

Для построения интерполяции-экстраполяции в *Mathcad* имеются несколько встроенных функций, позволяющих «соединить» точки выборки данных (x_i, y_i) кривой разной степени гладкости. По определению интерполяция означает построение функции $A(x)$, аппроксимирующей зависимость $y(x)$ в промежуточных точках (между x_i). Поэтому интерполяцию еще по-другому называют *аппроксимацией*. В точках x_i значения интерполяционной функции должны совпадать с исходными данными, т. е. $A(x_i) = y(x_i)$.

2.1.1. Линейная интерполяция

Самый простой вид интерполяции — линейная, которая представляет искомую зависимость $A(x)$ в виде ломаной линии. Интерполирующая функция $A(x)$ состоит из отрезков прямых, соединяющих точки (рис. 2.13).

Для построения линейной интерполяции служит встроенная функция *linterp* (рис. 2.13).

■ $linterp(x,y,t)$ — функция, аппроксимирующая данные векторов x и y кусочно-линейной зависимостью

- x — вектор действительных данных аргумента;
- y — вектор действительных данных значений того же размера;
- t — значение аргумента, при котором вычисляется интерполирующая функция.

Линейная интерполяция

$$x := (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6)^T$$

$$y := (4.1 \ 2.4 \ 3 \ 4.3 \ 3.6 \ 5.2 \ 5.9)^T$$

$$A(t) := linterp(x, y, t)$$

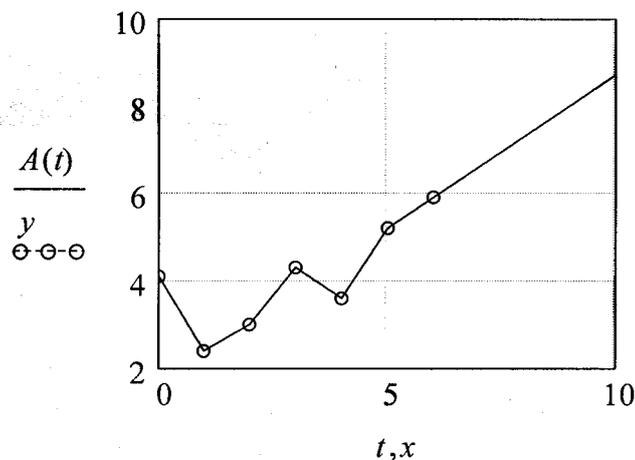


Рис. 2.13. Линейная интерполяция

Элементы вектора x должны быть определены в порядке возрастания, т.е. $x_1 < x_2 < x_3 < \dots < x_N$.

Как видно из рисунка, чтобы осуществить линейную интерполяцию, надо выполнить следующие действия:

1. Ввести векторы данных x и y (первые две строки рисунка).
2. Определить функцию $linterp(x, y, t)$.
3. Вычислить значения этой функции в требуемых точках.

2.1.2. Кубическая сплайн-интерполяция

Во многих практических задачах желательно соединить исходные точки не ломаной линией, а плавной кривой. Во многих случаях для этих целей подходит интерполяция кубическими сплайнами, т. е. отрезками кубических парабол (рис. 2.14).

Кубическая сплайн-интерполяция

$$x := (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6)^T$$

$$y := (4.1 \ 2.4 \ 3 \ 4.3 \ 3.6 \ 5.2 \ 5.9)^T$$

$$s := cspline(x, y) \quad A(t) := interp(s, x, y, t)$$

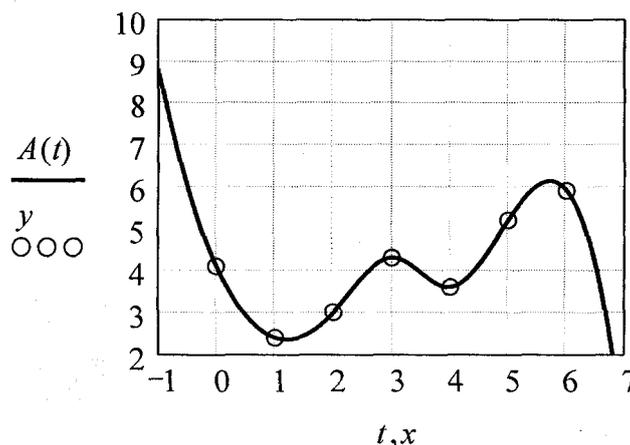


Рис. 2.14. Кубическая сплайн - интерполяция

■ $interp(s, x, y, t)$ — функция, аппроксимирующая данные векторов x и y кубическими сплайнами;

- s — вектор вторых производных, созданный одной из сопутствующих функций $cspline$, $pspline$ или $lspline$;

- x — вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;

- y — вектор действительных данных значений того же размера;

- t – значение аргумента, при котором вычисляется интерполирующая функция.

Сплайн-интерполяция в *Mathcad* реализуется несколько сложнее линейной. Перед применением функции *interp* необходимо вначале определить первый из ее аргументов – векторную переменную s . Делается это при помощи одной из трех встроенных функций тех же аргументов (x, y) :

- $lspline(x, y)$ — вектор значений коэффициентов линейного сплайна;

- $pspline(x, y)$ — вектор значений коэффициентов квадратичного сплайна;

- $cspline(x, y)$ — вектор значений коэффициентов кубического сплайна.

Выбор конкретной функции сплайновых коэффициентов влияет на интерполяцию вблизи конечных точек интервала (рис. 2.15)

Смысл сплайн – интерполяции заключается в том, что в промежутках между точками осуществляется аппроксимация в виде зависимости

$$A(t) := a \cdot t^1 + b \cdot t^2 + c \cdot t^3$$

Коэффициенты a, b, c, d рассчитываются независимо для каждого промежутка исходя из значений y_i в соседних точках.

Следует помнить, что если задать построение функции $A(x)$ вместо $A(t)$, то будет получено просто соединение исходных точек ломаной линией, так как в промежутках между точками вычисления интерполирующей функции не происходит.

Сплайн-интерполяция: линейная, квадратичная, кубическая и линейная интерполяция

$$x := (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6)^T$$

$$y := (4.1 \ 2.4 \ 3 \ 4.3 \ 3.6 \ 5.2 \ 5.9)^T$$

$$sl := lspline(x,y) \quad Al(t) := interp(sl,x,y,t)$$

$$sp := pspline(x,y) \quad Ap(t) := interp(sp,x,y,t)$$

$$s := cspline(x,y) \quad As(t) := interp(s,x,y,t)$$

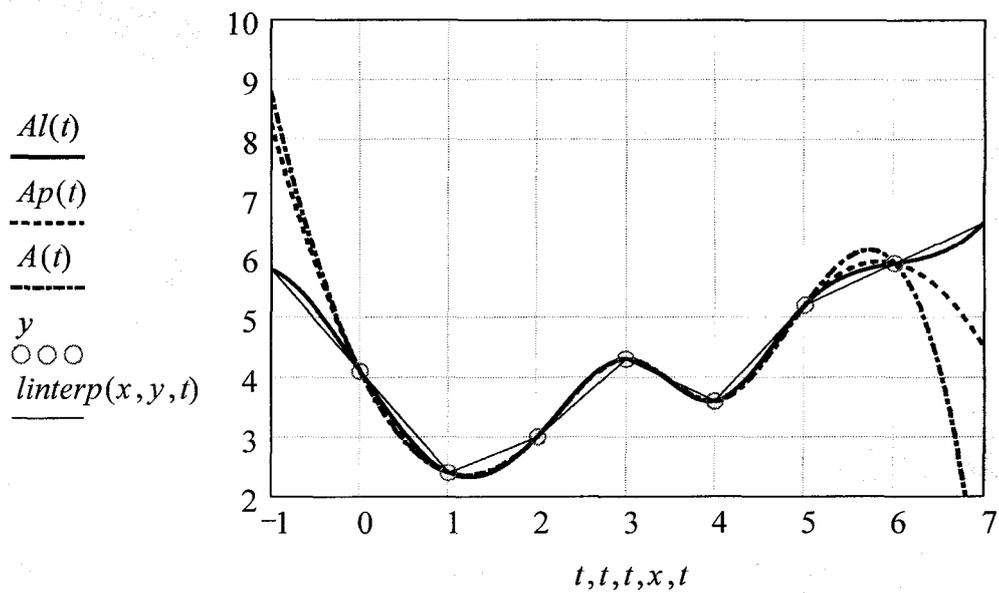


Рис. 2.15. Сравнение различных способов сплайн - интерполяции

ТЕМА 3. ВВОД-ВЫВОД ДАННЫХ

В данном разделе рассматриваются вопросы ввода входных данных в документы *Mathcad* и вывода результатов вычислений. Наиболее мощными средствами вывода результатов в *Mathcad* являются графики.

На применении динамической смены графиков основан аппарат создания видеофайлов анимации, который делает результаты работы в *Mathcad* особенно эффектными. К тому же, *Mathcad* обладает целым спектром возможностей по вводу-выводу данных во внешние текстовые и графические файлы.

3.1. Числовой ввод-вывод

Наиболее простой и распространенный ввод-вывод данных в *Mathcad* реализован присваиванием и (либо численным, либо символьным) выводом непосредственно в документе. Фактически документ *Mathcad* является одновременно и кодом программы и результатом ее выполнения. Поэтому самый простой и распространенный способ ввода-вывода — это непосредственное присвоение и вывод вычисленных значений в документах. Числовому вводу и выводу данных посвящен настоящий раздел. На рис. 3.1 приведены примеры для напоминания о форматах вывода числовых данных.

3.2. Создание графиков

В *Mathcad* встроено несколько различных типов графиков, которые можно разбить на две большие группы.

- Двумерные графики:

- XY (декартовый) график (*XY Plot*);
- полярный график (*PolarPlot*).

■ Трехмерные графики:

- график трехмерной поверхности (*Surface Plot*);
- график линий уровня (*Contour Plot*);
- трехмерная гистограмма (*3D Bar Plot*);
- трехмерное множество точек (*3D Scatter Plot*);
- векторное поле (*Vector Field Plot*).

Числовой ввод данных

$$i := 0..4 \quad x := 1.5257285 \quad y := 1234.567890 \quad f(x) := x^2$$

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Числовой вывод данных

$$x = 1.526 \quad f(x) = 2.328 \quad y = 1.235 \times 10^3 \quad y = 1234.568$$

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad f(A) = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$$

$i =$
0
1
2
3
4

$f(i) =$
0
1
4
9
16

Рис. 3.1. Примеры числовых ввода и вывода данных

Деление графиков на типы несколько условно, т.к., управляя установками многочисленных параметров, можно создавать комбинации типов графиков, а также новые типы (например, двумерная гистограмма распределения является разновидностью простого XY-графика).

Все графики создаются совершенно одинаково, с помощью панели инструментов **Graph** (График), различия обусловлены отображаемыми данными.

Чтобы создать график, например двумерный в декартовых координатах, следует:

1. Поместить курсор ввода в то место документа, куда требуется вставить график.

2. Если на экране нет панели **Graph** (График), то вызвать ее нажатием кнопки с изображением графиков на панели **Math** (Математика).

3. Нажать на панели **Graph** (График) кнопку **X-Y Plot** для создания двумерного графика или другую кнопку для иного желаемого типа графика.

4. После того, как в обозначенном месте документа появится пустая область графика с одним или несколькими местозаполнителями, следует ввести в местозаполнители имена переменных или функций, которые должны быть изображены на графике. В случае графика в декартовых координатах — это два местозаполнителя данных, откладываемых по осям x и y .

Если имена данных введены правильно, нужный график появится на экране. Созданный график можно изменить, в том числе меняя, сами данные, форматируя его внешний вид или добавляя дополнительные элементы оформления.

Самый наглядный способ создания графика — с помощью панели инструментов **Graph** (График). Однако точно так же создаются графики путем выбора соответствующего элемента подменю **Insert / Graph** (Вставка / График) либо нажатием соответствующей типу графика горячей клавиши.

Чтобы удалить график, нужно щелкнуть в его пределах и выбрать в верхнем меню *Edit* (Правка) пункт *Cut* (Вырезать) или *Delete* (Удалить).

3.3. Двумерные графики

К двумерным графикам относят графики в декартовой и полярной системах координат. Созданный график одного типа нельзя переделать в график другого типа (в отличие от трехмерных графиков). Для построения XU -графика необходимы два ряда данных, откладываемых по осям X и Y .

3.3.1. XU -график двух векторов

Самый простой и наглядный способ построить график в декартовых координатах — это сформировать два вектора данных, которые будут отложены вдоль осей X и Y . Последовательность построения графика двух векторов x и y показана на рис. 3.2. В этом случае в местозаполнители возле осей вводятся просто имена векторов. Также допускается откладывать по осям элементы векторов, т. е. вводить в местозаполнители возле осей имена x_i и y_i соответственно (рис. 3.3). В результате получается график, на котором отложены точки, соответствующие парам элементов векторов, соединенные отрезками прямых линий. Образованная ими ломаная называется рядом данных или кривой (*trace*).

Следует отметить, что подобным образом можно построить и XU – график столбцов или строк матриц, применяя, оператор выделения столбца и откладывая соответствующие выражения по осям графика.

3.3.2. XY–график вектора и ранжированной переменной

В качестве переменных, откладываемых на любой из осей, можно использовать саму ранжированную переменную (рис. 3.3). При этом на другой оси должно быть отложено либо выражение, явно содержащее саму ранжированную переменную, либо элемент вектора с индексом по этой ранжированной переменной, но никак не сам вектор.

3.3.3. XY–график функции

Нарисовать график любой скалярной функции $f(x)$ можно двумя способами. Первый заключается в дискретизации значений функции, присвоении этих значений вектору и прорисовке графика вектора. Собственно, так и были получены графики на рис. 3.3.

Второй, более простой способ, называемый *быстрым построением графика*, заключается во введении функции в один из местозаполнителей (например, y оси Y), а имени аргумента — в местозаполнитель x другой оси. В результате *Mathcad* сам создает график функции в пределах значений аргумента, по умолчанию принятых равными от -10 до 10 . Разумеется, впоследствии можно поменять диапазон значений аргумента, и график автоматически подстроится под него.

X_Y-график двух векторов;
 двух векторов, заданных элементами

$$i := 0..30 \quad x_i := i \cdot 0.2 \quad y_i := \cos(x_i) + 1$$

$$x^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2

$$y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	2	1.98	1.921	1.825	1.697	1.54	1.362	1.17	0.971	0.773	0.584

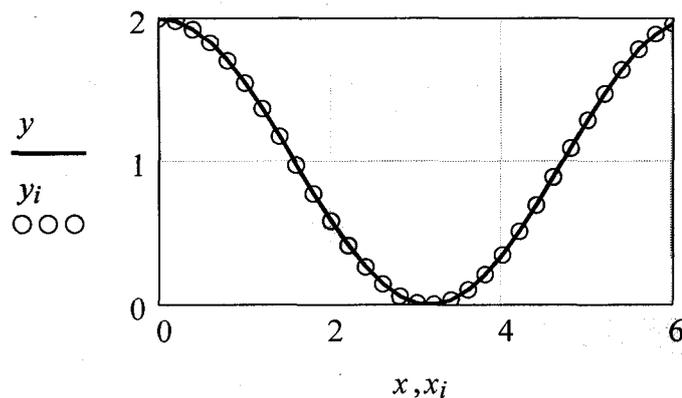


Рис. 3.2. График двух векторов x, y ;
 ○○○ график двух векторов, заданных элементами x_i, y_i

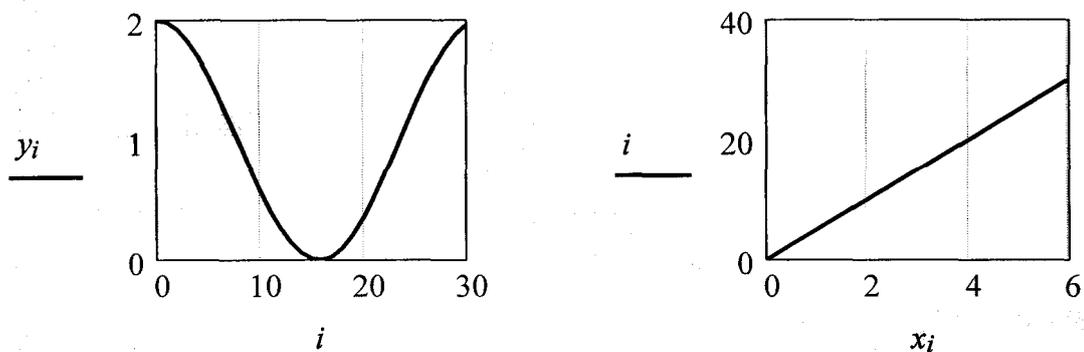


Рис. 3.3. График ранжированной переменной

Необходимо заметить, что если переменной аргумента функции было присвоено некоторое значение до построения в документе графика, то вместо быстрого построения графика будет нарисована зависимость функции с учетом этого значения. Примеры двух таких графиков приведены на рис. 3.4.

$i := 0..32$ $x_i := i \cdot 0.2$

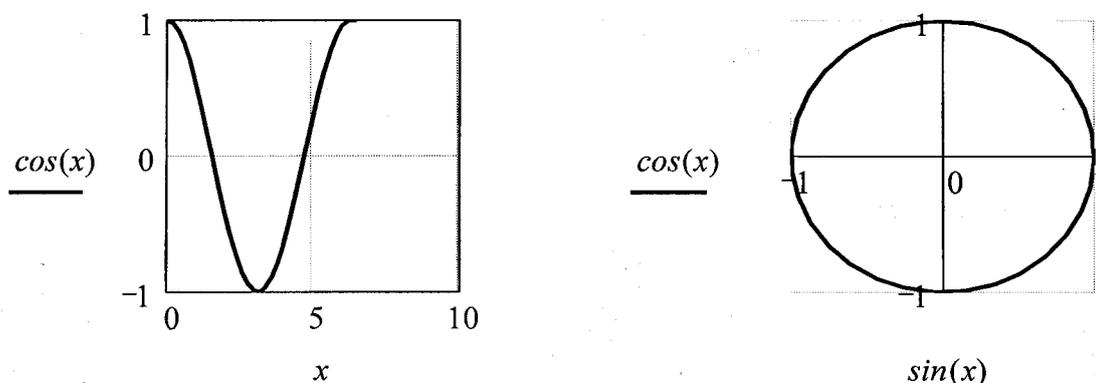


Рис. 3.4. Графики функций от векторного аргумента

3.3.4. Полярный график

Для создания полярного графика необходимо нажать кнопку **Polar Plot** на панели **Graph** (График) и вставить в местозаполнители имена переменных и функций, которые будут нарисованы в полярной системе координат: угол (нижний местозаполнитель) и радиус-вектор (левый местозаполнитель). Точно так же, как при создании графика в декартовых координатах, по осям могут быть отложены два вектора (рис. 3.5, слева), элементы векторов и ранжированные переменные в различных сочетаниях, а также может быть осуществлено быстрое построение графика функции (рис. 3.5, справа).

Форматирование полярных графиков практически идентично форматированию графиков в прямоугольных координатах, поэтому все, сказанное ниже об оформлении двумерных графиков на примере XY-графиков, в полной мере относится и к полярным.

$i := 0..10$ $x_i := i \cdot 0.4$ $z := 0,0.01..2 \cdot \pi$

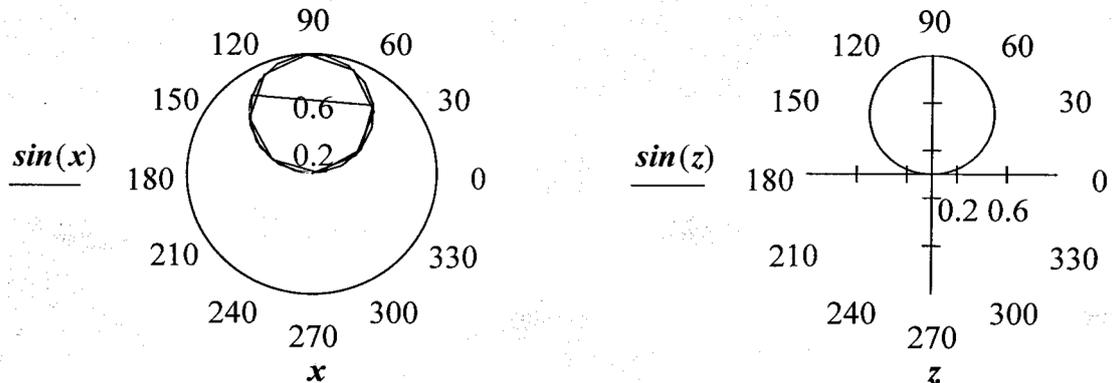


Рис. 3. 5. Полярные графики

3.3.5. Построение нескольких рядов данных

На одном графике может быть отложено до 16 различных зависимостей. Чтобы построить на графике еще одну кривую, необходимо выполнить следующие действия:

1. Поместить линии ввода таким образом, чтобы они целиком захватывали выражение, стоящее в надписи координатной оси Y (рис. 3.5).
2. Нажать клавишу \langle, \rangle .
3. В результате появится местозаполнитель, в который нужно ввести выражение для второй кривой.
4. Щелкнуть в любом месте вне этого выражения (на графике или вне его).

После этого вторая кривая будет отображена на графике. На рис.3.5 уже нарисованы два ряда данных, а нажатие клавиши с запятой \langle, \rangle приведет к появлению третьего местозаполнителя, с помощью которого можно определить третий ряд данных.

$$i := 0..50 \quad x_i := i \cdot 0.2 \quad y_i := \sin(x_i)$$

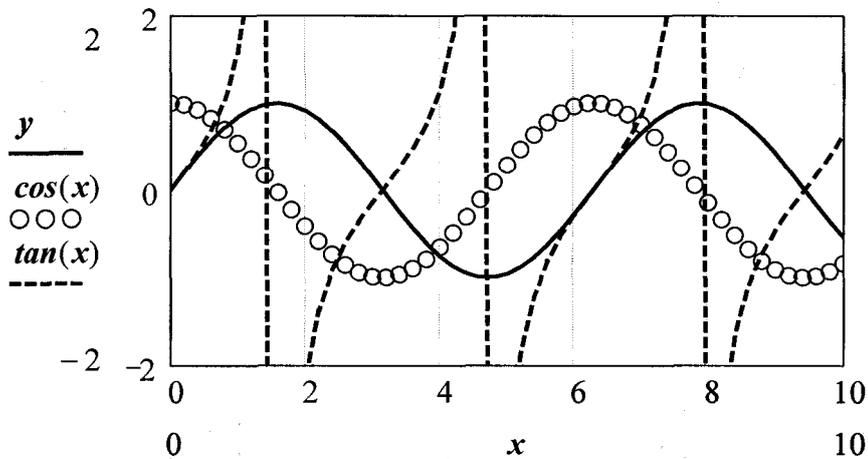


Рис. 3.5. Построение нескольких графиков от одного аргумента

Описанным способом будет создано несколько зависимостей, относящихся к одному аргументу. На рис. 3.6. построены графики пар точек $y(x_i)$ и $\cos(x_i)$ одного и того же аргумента — элементов вектора x_i . Об этом говорит единственная метка x у оси абсцисс. Вместе с тем, имеется возможность отображения на одном и том же графике зависимостей разных аргументов. Для этого достаточно расставить по порядку метки всех зависимостей у обеих осей.

3.4. Ввод – вывод во внешние файлы

Важный компонент ввода–вывода — это ввод – вывод во внешние файлы. Ввод внешних данных в документы *Mathcad* применяется чаще вывода, поскольку *Mathcad* имеет гораздо лучшие возможности представления результатов расчетов, чем многие пользовательские программы. Для общения с внешними файлами данных в *Mathcad* имеется несколько разных способов. Самый простой из них — использовать имеющееся семейство встроенных функций.

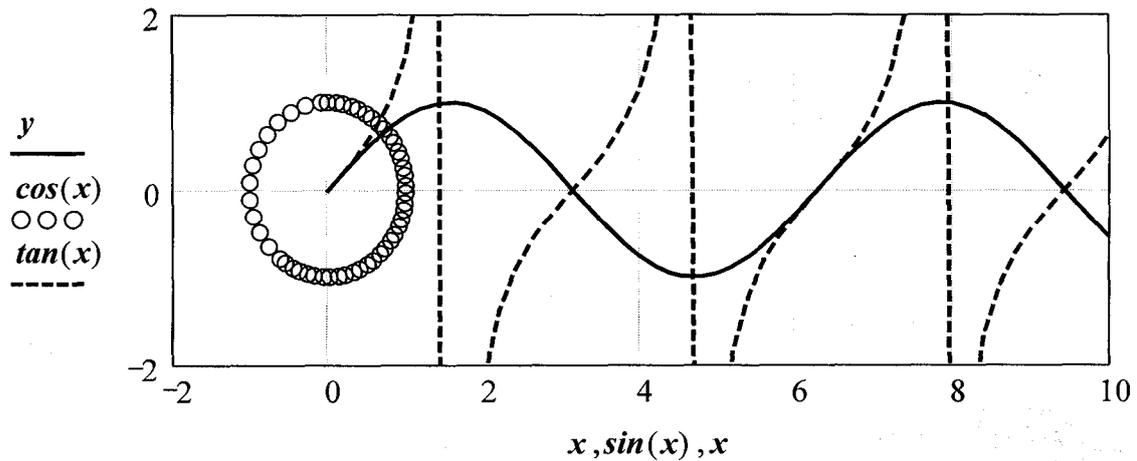


Рис. 3.6. Построение нескольких графиков от разных аргументов

3.4.1. Текстовые файлы

В *Mathcad* имеются встроенные функции для работы с текстовыми файлами (рис. 3.7, рис. 3.8):

- $READPRN("file")$ – чтение данных в матрицу из текстового файла;
- $WRITEPRN("file")$ – запись данных в текстовый файл;
- $APPENDPRN("file")$ – дозапись данных в существующий текстовый файл;

- $file$ – путь к файлу.

Можно задавать как полный путь к файлу, так и относительный, он будет отсчитываться от папки, в которой находится файл с документом *Mathcad*.

Запись матрицы **M1** в текстовый файл

$$M1 := \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 9 & 8 & 7 & 6 \\ 7 & 6 & 5 & 4 & 3 \\ 2 & 1 & 1 & 2 & 3 \end{pmatrix} \quad \text{WRITEPRN("fileM1.prn")} := M1$$

Считывание данных из текстового файла в матрицу **M2**

$$M2 := \text{READPRN("fileM1.prn")}$$

$$M2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 9 & 8 & 7 & 6 \\ 7 & 6 & 5 & 4 & 3 \\ 2 & 1 & 1 & 2 & 3 \end{pmatrix}$$

Дозапись вектора **V1** в существующий текстовый файл **fileM1**

$$V1 := (9 \ 9 \ 9 \ 9 \ 9)$$

$$\text{APPENDPRN("fileM1.prn")} := V1$$

Рис. 3.7. Запись данных матрицы в текстовый файл, считывание текстового файла в матрицу, дозапись данных в существующий текстовый структурированный файл

Следует помнить, что если выводятся данные в файл встроенной функцией *WRITEPRN*, то в любом случае создается новый текстовый файл. Если до записи данных файл с таким именем существовал, то его содержимое будет уничтожено заменой новыми данными.

Если необходимо сохранить прежнее содержимое текстового файла с данными, то нужно воспользоваться функцией *APPENDPRN*. Эта же встроенная функция может применяться и для создания нового файла. Иными словами, если файла с заданным именем не существовало, то он, после применения, будет создан и наполнен теми данными, которые определены

в документе.

Файл, созданный функцией WRITEPRN

1	2	3	4	5
0	9	8	7	6
7	6	5	4	3
2	1	1	2	3
9	9	9	9	9

Файл, созданный функцией APPENDPRN

1	2	3	4	5
0	9	8	7	6
7	6	5	4	3
2	1	1	2	3
9	9	9	9	9
9	9	9	9	9

Рис. 3.8. Форматы файлов, созданных функциями *WRITEPRN*, *APPENDPRN*

3.4.2. Графические файлы

Подобно вводу–выводу в текстовые файлы можно организовать чтение и запись данных в графические файлы различных форматов.

- *READRGB*("file") – чтение цветного изображения;
- *READBMP*("file") – чтение изображения в оттенках серого;
- *WRITERGB*("file") – запись цветного изображения;
- *WRITEBMP*("file") – запись изображения в оттенках серого;
- *file* – путь к файлу.

Имеется также большое количество функций специального доступа к графическим файлам, информацию о которых можно найти, например, в справочной системе *Mathcad*.

3.5. Ссылки и гиперссылки

Гиперссылки — это активные области в документах *Mathcad*, которые выводят на экран какое-либо другое место в активном документе:

другой документ *Mathcad* или другое приложение, либо на сайт в Интернете. Гиперссылки эффективны в больших документах, а также в обучающих и презентационных системах, выполненных в *Mathcad*. Особенно важно уметь пользоваться гиперссылками, если разрабатываются электронные книги.

3.5.1. Установка тега

Прежде чем определить гиперссылку, нужно сначала точно определить место в документе, на которое эта гиперссылка будет переводить курсор, и которое в *Mathcad* называется *тегом (tag)*. Для установки тега нужно:

1. Щелкнуть на том месте, где необходимо расположить тег, правой кнопкой мыши.
2. В контекстном меню выбрать пункт ***Properties*** (Свойства).
3. В диалоге ***Properties*** (Свойства) перейти на вкладку ***Display*** (Отображение).
4. В поле ***Tag*** (Тег) ввести имя тега, которое будет идентифицировать данное место в документе.
5. Нажать кнопку **ОК**.

3.5.2. Вставка гиперссылки

Создавать гиперссылку можно в любом месте любого документа. Щелчок на гиперссылке будет переводить курсор на место тега, на который установлена гиперссылка, либо (при отсутствии тега) в начало документа, на который произведена гиперссылка. Для вставки гиперссылки следует:

1. Щелкнуть на текстовой или формульной области документа, которую предполагается сделать гиперссылкой.
2. Выбрать в меню ***Insert*** (Вставка) пункт ***HyperLink*** (Гиперссылка).

3. В диалоговом окне *Edit HyperLink* (Правка гиперссылки), в текстовом поле *Link to file or URL* (Связать с файлом или *Universal Resource Locator*, универсальный адрес ресурсов)) определить путь к документу, на который нужно ссылаться, а также (необязательно) имя тега (*tagename*) с указанием имени файла (*filename*) в формате *name tagename*.

4. По желанию, в нижнем текстовом поле можно задать текст, который будет появляться на строке состояния при наведении указателя мыши на гиперссылку.

5. Нажать кнопку **ОК**.

Следует помнить, что имя файла необходимо указывать, даже если тег расположен в том же документе, что и гиперссылка.

Если Вы все сделали правильно, то при двойном щелчке на гиперссылке будет осуществлен переход на место, где расположен тег.

Чтобы отредактировать гиперссылку, достаточно, находясь на ее области, выбрать тот же пункт меню *Insert / HyperLink* (Вставка / Гиперссылка). Появится диалог *Edit HyperLink* (Правка гиперссылки), в котором, можно исправить ее параметры.

Удалить гиперссылку можно нажатием кнопки *Remove Link* (Удалить гиперссылку).

Помимо гиперссылок на документы *Mathcad* допускается создавать гиперссылки на другие файлы (например, видеофайлы или *HTML*-файлы), в том числе, находящиеся в Интернете. Для этого достаточно указать соответствующий адрес *URL* в верхнем текстовом поле диалога *Edit HyperLink* (Вставка гиперссылки).

3.5.3. Ссылки

Помимо гиперссылок иногда стоит применять другие, схожие с ними объекты, называемые *ссылками (Reference)*. Ссылка на документ *A*,

вставленная в некоторое место документа *B*, приводит к расчету всего документа *A* внутри документа *B*. Таким образом, ссылки позволяют хранить вложенные друг в друга расчеты в разных файлах.

Практика показала, что ссылки могут быть необходимым инструментом при разработке систем автоматизированного проектирования, расчета или пакета прикладных программ (ППП). В этом случае следует разработать соглашения об именах глобальных и локальных переменных, которые будут использоваться в программных документах.

Для установки ссылки достаточно выбрать команду *Insert / Reference* (Вставка / Ссылка) и затем в диалоговом окне *Insert Reference* (Вставка ссылки) определить путь к имени файла-ссылки. После нажатия кнопки ОК на месте курсора ввода появится информация о файле-ссылке, а результат будет рассчитан в соответствии с формулами этого файла.

ТЕМА 4. СИСТЕМЫ АВТОМАТИЗИРОВАННОГО РАСЧЕТА

4.1. Понятие систем автоматизированного расчета (САР)

Программные средства для решения различных типов вычислительных задач можно разделить на 4 группы:

- отдельные прикладные программы;
- библиотеки прикладных программ;
- пакеты прикладных программ;
- интегрированные программные системы.

Отдельная прикладная программа пишется, как правило, для решения конкретной прикладной задачи. Примерами могут служить программа решения системы линейных алгебраических уравнений тем или иным численным методом, программа вычисления собственных значений матрицы и т. д. Разработчиками таких программ являются прикладные программисты, специализирующиеся в соответствующих предметных областях. Прикладная программа может быть реализована в виде набора модулей, каждый из которых выполняет некоторую самостоятельную функцию. Вызов прикладной программы производится управляющей программой с использованием штатных средств. Непосредственное использование прикладной программы предполагает наличие определенной программистской квалификации пользователя: знание языка программирования и особенностей его конкретной версии, умение работать в рамках соответствующей системы программирования.

Необходимость решения некоторого множества задач предметной области (или нескольких предметных областей), а также обеспечение возможности применения, различных методов при решении той или иной конкретной задачи привело к появлению библиотек программ.

Библиотека представляет собой набор отдельных программ, каждая из которых решает некоторую прикладную задачу. Библиотеки программ зарекомендовали себя эффективным средством решения вычислительных задач. Они интенсивно используются при решении научных и инженерных задач с помощью ПЭВМ. Условно их можно разделить на библиотеки широкого применения и специализированные библиотеки.

Программы библиотеки широкого применения, предназначены для решения задач из различных предметных областей. Примером служат программы интерполяции, аппроксимации, решения задач линейной алгебры, встречающиеся во многих вычислительных задачах, и т.д.

Специализированные библиотеки ориентированы на решение отдельных, порой достаточно узких, классов задач. В качестве примера можно привести комплекс (блок программ) для автоматизированного профилирования лопаток турбин [6], комплекс программ [7] для расчета цилиндрической камеры сгорания газотурбинного двигателя (ГТД) и т.д.

Переход от разработки отдельных прикладных программ к созданию библиотек программ ставит перед разработчиками проблемы систематизации, документирования, тестирования и переносимости (этим вопросам посвящена работа [8]).

Проблема систематизации состоит в разбиении библиотеки на разделы и подразделы в соответствии с классификацией задач предметной области и методов их решения. Проблема документирования заключается в составлении единых правил описания программ библиотеки.

Тестирование библиотеки состоит в проверке программ на специально подготовленных тестовых данных: определение области применимости программы, выявление разнообразных количественных характеристик программы (скорость работы, точность получаемых результатов и т. п.), сравнение с другими программами для решения той же задачи.

Проблема переносимости состоит в разработке методов и средств, обеспечивающих возможность использования программ библиотеки в различных вычислительных условиях (на различных типах ЭВМ, в различных операционных системах и т.д.). Доступ к программам библиотеки осуществляется с использованием штатных средств того или иного языка программирования. Таким образом, как и при работе с отдельными прикладными программами, от пользователя требуется определенная квалификация в области программного обеспечения ПЭВМ и умение программировать.

Разработка библиотек программ обычно осуществляется силами прикладных программистов. При этом нередко в библиотеку включаются программы, написанные в разное время. Как правило, библиотеки программ ориентированы на типовые задачи предметной области.

Рассмотрим пакет прикладных программ (ППП) как самостоятельную форму прикладного программного обеспечения. Для этого необходимо уточнить само понятие пакета.

В настоящее время не существует признанной всеми специалистами единой точки зрения по этому вопросу.

Так, в работе [9] ППП определяется как совокупность программ для решения определенного класса задач. Для решения конкретной задачи с помощью ППП пользователь должен задать определенную информацию, например, указать последовательность выполнения программ пакета и режимы использования отдельных программ.

В статье [10] ППП определяется как совокупность программ, совместимых по структуре данных, способам управления, объединяемых общностью функционального назначения и представляющих собой средство решения класса задач определенным кругом пользователей. При этом под классом задач понимается множество прикладных проблем, обладающих общностью применяемых алгоритмов и информационных массивов.

В книге [11] дается определение пакета как комплекса взаимосвязанных программ, обладающих специальной организацией, которая обеспечивает значительное повышение производительности труда программистов и пользователей пакета. В данном случае не делается попытки выделить ППП среди других форм программного обеспечения ЭВМ.

В статье [12] под ППП понимается комплекс взаимосвязанных прикладных и системных программ, обеспечивающих адекватное покрытие некоторой прикладной деятельности. При этом подчеркивается необходимость наличия в составе пакета системных средств, а также связи между прикладными и системными программами пакета. Это определение исключает из числа ППП простые тематические подборки программ, а также библиотеки программ, поскольку они не содержат достаточно развитых системных компонентов.

В книге [13] даются два различных определения ППП, одно из которых ближе прикладным программистам, а другое - системным:

1. ППП – средство автоматизации решения класса прикладных задач;
2. ППП – форма существования проблемно-ориентированного программного обеспечения.

При этом отмечается, что эти определения являются эквивалентными и попытки дополнить или формализовать определение пакета приводят к спорным утверждениям, не увеличивая информативности.

Очевидно, что любое формальное определение ППП либо не является однозначным и не характеризует достаточно четко множество программных систем, называемых пакетами (прежде всего по причине неоднозначности и неопределенности используемых терминов), либо необоснованно исключает из числа пакетов отдельные типы программных систем.

Будем считать пакетом программ любой комплекс, ориентированный на решение некоторого класса задач. Формально такое определение не исключает из числа пакетов и библиотеки программ. Однако можно указать на ряд характерных отличительных особенностей пакетов.

Одной из главных особенностей является *ориентация* ППП не на отдельную задачу, а на некоторый класс задач, включающий специфические задачи предметной области. Отсюда следует необходимость модульной организации ППП как основного технологического принципа его конструирования.

Суть этого принципа состоит в оформлении общих фрагментов используемых алгоритмов в виде самостоятельных модулей. Решение сформулированной пользователем задачи осуществляется некоторой цепочкой таких модулей. Организация же пакета по принципу библиотеки (т.е. в виде совокупности программ, каждая из которых решает содержательную прикладную задачу) во многих случаях оказывается неэффективной из-за наличия значительного числа общих частей в алгоритмах решения различных задач одного класса.

Другой особенностью ППП является наличие в его составе специализированных языковых средств, обеспечивающих удобную работу пользователя с пакетом. Еще одна особенность ППП состоит в наличии специальных системных средств, обеспечивающих средства информационного обеспечения, средства взаимодействия пакета с операционной системой и т. п.

Наконец, интегрированной программной системой назовем комплекс программ, элементами которого являются различные пакеты и библиотеки программ. Примером служат системы автоматизированного проектирования (САПР), системы автоматизированного расчета (САР), имеющие в своем составе несколько ППП различного назначения. Часто в

подобной системе решаются задачи, относящиеся к различным классам или даже к различным предметным областям.

Следует указать на отсутствие четких и однозначных границ между перечисленными формами прикладного программного обеспечения. Переход от создания библиотек программ к разработке систем был вызван целым рядом причин. К их числу, прежде всего, относится резкое увеличение возможностей ЭВМ (ПЭВМ).

4. 2. Структура ППП (САР) и основные компоненты

Несмотря на большое разнообразие конкретных пакетных разработок и САР, можно выделить следующие основные компоненты:

- входные языки;
- предметное обеспечение;
- системное обеспечение.

Важно отметить, что такое разбиение на составные элементы отражает в первую очередь функции, выполняемые программами систем, а не структуру самих программ, которая зависит от индивидуальных особенностей конкретного пакета. В разных пакетах указанные компоненты могут быть развиты в различной степени или вовсе отсутствовать. Однако наиболее развитые ППП часто обладают всеми этими компонентами, каждый из которых может иметь довольно сложную структуру.

Входные языки являются средством общения пользователя с пакетом. Как отмечалось ранее, развитый пакет может обладать несколькими входными языками, предназначенными для выполнения различных функций и ориентированными на различные типы пользователей.

Можно выделить следующие основные типы пользователей САР:

– разработчик ППП (САР), осуществляющий его модификацию и развитие с учетом изменения круга пользователей, класса решаемых задач (появление новых типов задач, развитие численных методов, модификация форм проведения работ и т. д.), а также состава аппаратного и программного обеспечения ПЭВМ;

– ответственный за сопровождение, в функции которого входит поддержание пакета в работоспособном состоянии в условиях конкретной вычислительной системы (обеспечение сохранности программ и массивов данных, своевременное дублирование информационных файлов, выявление ошибок в программах пакета);

– администратор, отвечающий за организацию доступа пользователей к пакету, содержимое базы данных, защиту информации от несанкционированного доступа;

– конечный пользователь, применяющий пакет для решения конкретных прикладных задач.

В качестве входных языков могут использоваться как универсальные, так и специализированные языки программирования. Например, в качестве входного языка разработчика САР для написания прикладных и системных программ пакета обычно используется тот или иной универсальный язык программирования (Фортран, Паскаль и т.п.). В то же время входной язык конечного пользователя в развитом пакете, как правило, является языком качественно более высокого уровня по сравнению с универсальными языками. Изобразительные средства такого языка учитывают особенности задач предметной области и специфику пользователей. Подобные языки называют проблемно-ориентированными, или предметно-ориентированными.

Следующий компонент ППП (САР) – конкретная прикладная деятельность, которая характеризуется двумя факторами:

1. Классом решаемых задач и используемых для этих целей методов;

2. Дисциплиной работы, т. е. совокупностью правил, соглашений и технологических приемов, принятых при разработке, отладке и эксплуатации программ.

Предметное обеспечение представляет собой компонент пакета, отражающий особенности первого из этих факторов, т.е. особенности конкретной предметной области. Предметное обеспечение включает:

– программные модули, реализующие алгоритмы (или их отдельные фрагменты) решения прикладных задач;

– средства сборки программ из отдельных модулей.

Определение состава библиотеки модулей и форм их взаимодействия между собой является одной из наиболее трудоемких задач при построении ППП. Ее решение предполагает проведение тщательного и квалифицированного модульного анализа используемых алгоритмов. Удачно проведенный модульный анализ в значительной степени влияет на полноту охвата предметной области, а также на возможность расширения класса решаемых задач. Таким образом, выделение модулей существенно зависит от специфики задач и используемых алгоритмов и, по сути дела, отражает принятый в пакете способ сборки программ.

Наиболее распространено в настоящее время оформление каждого модуля в виде программной единицы на том или ином языке программирования (например, в виде подпрограммы или подпрограммы-функции. Такой модуль обеспечивает решение некоторой самостоятельной задачи и связан с другими модулями лишь входной и выходной информацией. Организация предметного обеспечения в виде библиотеки программ характерна для большинства существующих ППП (САР).

Помимо рассмотренного подхода к оформлению модулей как программных единиц используются и другие способы.

Для сборки _программ из модулей могут использоваться различные средства: от ручной сборки, выполняемой самим пользователем, до полностью автоматической.

Ручная сборка заключается в явном указании пользователем последовательности выполняемых модулей. При этом могут использоваться различные способы организации взаимодействия модулей, т.е. различные способы передачи параметров между модулями, выполнения необходимых промежуточных преобразований данных управления памятью и т. д. Если такое взаимодействие обеспечивается самим пользователем, то работа с пакетом по существу не отличается от работы с библиотекой программ. Однако даже при ручной сборке пользователю могут предоставляться средства автоматизации этих процессов.

Системное обеспечение представляет собой совокупность системных средств (программы, файлы, таблицы и т.д.), обеспечивающих определенную дисциплину работы пользователя при решении прикладных задач. Системное обеспечение в ряде ППП и САР по существу является специализированной операционной системой, определяющей операционное окружение пакета. В заключение отметим, что создание ППП(САР) как наиболее развитой на сегодняшний день формы организации прикладного программного обеспечения предполагает решение всех тех проблем, которые возникают при разработке библиотек программ. К ним, в частности, относятся проблемы тестирования как пакета в целом, так и его отдельных компонентов, переносимости ППП из одной вычислительной среды в другую, автоматизации документирования пакетов и т.д.

4. 3. Показатели качества ППП (САР)

Разработка методики оценки качества пакетов и систем имеет важное значение для эффективности их разработки и эксплуатации. Главная

трудность разработки методики, устанавливающей общие правила оценки научно-технического уровня и качества программной продукции, состоит в огромном разнообразии применений ПЭВМ, что, в свою очередь, порождает разнообразие критериев качества программных изделий в зависимости от их назначения и особенностей той или иной предметной области. Кроме того, оценка качества программного изделия во многом зависит от требований, предъявляемых пользователями, поскольку основным критерием хорошей программы является адекватное соответствие запросам конкретного пользователя. Еще одна трудность, возникающая при создании такой методики, заключается в том, что характеристики качества программ (эффективность, надежность, простота использования, степень переносимости и т. д.) часто не поддаются количественной оценке. Для оценки конкретной САР могут использоваться весьма разнородные показатели, каждый из которых отражает то или иное свойство пакета. Степень важности отдельного показателя зависит от типа пользователей, особенностей операционного окружения пакета и целого ряда других параметров. Некоторые из этих показателей являются взаимно противоречивыми в том смысле, что улучшение одних показателей влечет за собой ухудшение других. Например, надежность пакета и время, затраченное на его разработку.

В настоящее время отсутствуют какие-либо общие рецепты определения значимости отдельных показателей при оценке качества конкретного пакета. Эти вопросы в большинстве случаев решаются на основе накопленного опыта.

Можно выделить две группы показателей качества ППП.

К первой относятся показатели, отражающие функциональные возможности САР и имеющие важное значение для пользователей.

Вторую группу составляют показатели, характеризующие структурные свойства системы; эти показатели существенны, в первую очередь, для разработчиков и влияют на простоту сопровождения и модификации ППП. Рассмотрим показатели каждой из этих групп.

К показателям качества функциональных возможностей вычислительных ППП в первую очередь относятся: точность результатов, время работы, используемые ресурсы ЭВМ (память различных уровней, внешние устройства и др). Эти показатели наиболее просто формализуются и чаще всего используются при сравнении различных пакетов.

Точность работы пакета характеризуется расхождением между точным математическим результатом и результатом, полученным в ходе вычислений. Определение разумной меры точности может оказаться относительно несложной проблемой в случаях, когда сама задача допускает достаточно простую математическую постановку. Однако для некоторых задач (например, для задач вычислительного эксперимента) определение меры точности является непростой проблемой, связанной с физической интерпретацией полученных результатов.

Определение такого показателя, как время работы ППП, обычно не вызывает затруднений. Однако в ряде случаев возможна очень сложная зависимость времени счета от входных данных. Если в процессе своей работы программы пакета обращаются к другим программам или внешним устройствам, то более интересной и важной может оказаться характеристика ППП – количество подобных обращений к внешним объектам.

К ним относятся следующие показатели удобства использования ППП:

– возможность возникновения аварийных ситуаций в процессе работы пакета;

– квалификация пользователя, необходимая для полноценной эксплуатации пакета;

– степень пригодности пакета для решения некоторого заданного класса задач;

– гарантии разработчиков, предоставляемые пользователям.

Если аварийная ситуация возникает в результате неправильно заданных пользователем входных данных (например, входные данные выходят за рамки области допустимых значений). Однако с точки зрения удобства эксплуатации САР при неверно заданных входных данных более целесообразным представляется не выход на аварийную ситуацию, а выдача пользователю соответствующего диагностического сообщения. Другой важный показатель качества ППП связан с квалификацией пользователя, необходимой для эффективной работы с пакетом. Так, для менее квалифицированного пользователя наиболее подходящим может оказаться режим меню, при котором активной стороной диалога является пакет. Достаточно квалифицированный пользователь, уже имеющий опыт работы с ППП, может предпочесть командный режим, предоставляющий широкие возможности выбора конкретных действий.

Некоторые из рассмотренных показателей являются противоречивыми. Анализ процессов разработки, сопровождения и модернизации ППП приводит к необходимости использования самостоятельной группы показателей качества. Эти показатели отражают, прежде всего, строгое следование на указанных этапах жизненного цикла ППП некоторой системе стандартов.

Наличие стандартов – неотъемлемый элемент любой промышленной технологии. По этой причине особое значение вопросы стандартизации имеют сегодня, когда производство программ и их документации приобретает промышленный характер. Введение стандартов преследует следующие основные цели:

– упрощение процессов разработки и эксплуатации программ и, следовательно, повышение их надежности и уменьшение стоимости;

– облегчение чтения и понимания программ, что в свою очередь упрощает их тестирование, позволяет снизить число ошибок, облегчает модернизацию.

Перечислим некоторые другие объекты стандартизации в программах пакета:

- способы представления данных в памяти ЭВМ;
- диагностика аварийных ситуаций, возникающих на различных этапах решения задачи;
- правила наименования программ и используемых объектов данных (система наименований должна создаваться в соответствии с некоторой классификацией задач предметной области и допускать модификацию и развитие пакета);
- соглашения по использованию констант, в частности машинно-зависимых констант, значения которых определяются величиной разрядной сетки конкретной ЭВМ;
- соглашения по работе с внешними устройствами, в том числе правила использования операторов ввода/вывода;
- схема управления памятью различных уровней, в том числе оперативной;
- соглашения по использованию внешних программ, к числу которых относятся программы пользователя, различные компоненты операционной системы и системы программирования, отдельные служебные программы и т. д.

Наличие таких стандартов существенно облегчает обмен программными изделиями и применение ранее созданных программ в новых разработках. Кроме того, появляется возможность автоматизировать подготовку документации путем использования специализированных инструментальных средств.

4. 4. Разработка и оформление модулей

Модульный принцип является основным технологическим принципом конструирования современных ППП (САР). Его суть заключается в выделении множества независимых элементов (модулей) пакета, из которых осуществляется сборка целевой программы для решения сформулированной пользователем задачи. При этом независимость модулей означает возможность их независимой разработки и использования. Определение структуры и свойств модуля зависит от специфики конкретного ППП или системы. Это объясняется тем, что различные типы пакетов могут иметь принципиально различную организацию, определяемую особенностями функционального назначения, методами управления решением задач, формами представления и назначением результатов, а также другими параметрами. Таким образом, сформулировать свойства модуля как конструктивного элемента пакета возможно лишь применительно к конкретному пакету и по существу требования к модулю характеризуют используемый в пакете метод сборки программ.

При разработке модулей необходимо учитывать различные, порой противоречивые, критерии. Рассмотрим два важных свойства модуля: актуальность и содержательность (см. [15]).

Актуальность модуля определяется числом различных контекстов его применения. *Содержательность* характеризует пригодность модуля для решения некоторой нетривиальной задачи. Указанные требования к модулю в принципе являются противоречивыми. Действительно, если исходить лишь из требования актуальности, то модулями следует объявить отдельные машинные команды или операторы языка программирования. Удовлетворение же лишь второму требованию приведет к тому, что в качестве модулей будут использоваться программы решения самостоятель-

ных прикладных задач. Выход из этой ситуации заключается в поиске компромиссного решения, при котором достигается максимальное значение отношения K/N , где N – общее количество модулей пакета; K – число различных типов решаемых пакетом задач.

Указанный критерий лишь частично характеризует эффективность модульного разбиения и не затрагивает многих других важных аспектов. В число прочих критериев качества модульной структуры входят эффективность сборки целевой программы из отдельных модулей (быстрота сборки и требуемые ресурсы ПЭВМ), полнота охвата предметной области, простота модификации и расширения пакета, возможность использования уже готовых программ и др.

Основой разработки библиотеки модулей является модульный анализ, который включает в себя несколько этапов. Схема процесса разработки прикладной программы приведена на рис. 4.1.

Приведенная технологическая цепочка имеет циклическую структуру. При этом вход в цикл возможен в любом звене цепочки, однако точка выхода является единственной и ее достижение предполагает прохождение всех промежуточных звеньев.

На этапе физической постановки задачи модульный анализ выполняется с целью выделения самостоятельных задач, которые не пересекаются по своему физическому содержанию. Результатом является совокупность физических модулей и схема связей между ними по информации и управлению. Под физическим модулем здесь понимается общепринятое определение физической задачи. Например, требуется определить скорость, температуру и давление идеального газа при неустановившемся течении его вдоль длинной теплоизолированной трубы переменного сечения с заданными условиями на ее концах. Такая формулировка не дает точной информации, но предполагает известными геометрические параметры тру-

бы, условия регулирования на концах, уравнение состояния газа и силы взаимодействия газа с твердой стенкой.

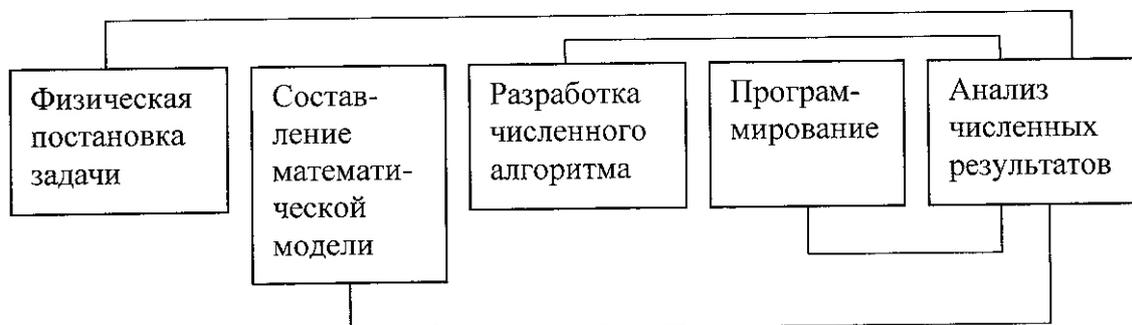


Рис. 4.1. Этапы решения физических задач

На втором этапе осуществляется математическая постановка задачи: выписываются основные уравнения и условия, при которых они решаются. Проводится теоретический анализ математической модели. При этом может получиться, что разные физические модули имеют одну и ту же математическую формулировку. В результате данного этапа создается совокупность математических модулей, которые описывают изучаемый физический процесс.

На третьем этапе разрабатываются численные алгоритмы, реализуемые модулями. Цель этого этапа модульного анализа заключается в построении совокупности неповторяющихся алгоритмических модулей, позволяющих сконструировать численный алгоритм реализации каждого математического модуля.

На этапе программирования (четвертый этап) модульный анализ направлен на выделение общих программных частей алгоритмов. Результатом является разработка программных модулей, обеспечивающих решение задачи моделирования соответствующего физического процесса.

Существуют некоторые другие характеристики модулей, которые также необходимо учитывать при определении модульной структуры пакета.

Одной из характеристик является количество входов и выходов модуля. В большинстве случаев наилучшим решением является наличие одного входа и одного выхода. Это упрощает организацию взаимодействия модулей и облегчает сопровождение пакета несмотря на то, что программа с несколькими входами и выходами может оказаться короче и занимать меньше места в памяти.

Другой характеристикой модуля является его объем. Применение небольших модулей имеет определенные преимущества (к числу больших можно отнести модули, состоящие из нескольких сотен строк на том или ином языке программирования). Небольшие модули позволяют строить программы, которые легче редактировать и модернизировать, такие модули легче и качественнее тестируются. С другой стороны, небольшие модули дольше проектируются, требуют большего числа связей, медленнее работают, требуют большего объема документации.

Сложность модуля должна быть ограничена, поскольку именно ограничение сложности и является основой модульного программирования. Один из путей удовлетворения указанному требованию заключается в ограничении объема модуля. Другое ограничение состоит в том, что каждый модуль должен выполнять только одну функцию, причем целиком. Концепция «один модуль – одна функция» служит ключом к хорошо спроектированным программам. Другими словами, каждый модуль должен решать некоторую небольшую, но самостоятельную задачу. Примерами таких задач являются: вычисление обратной матрицы, редактирование запроса, загрузка главного файла, печать итоговой таблицы и т.д. К числу других параметров, которые могут подвергаться ограничениям,

относятся: размер используемой оперативной памяти, число возможных ветвлений в программе модуля, время разработки модуля.

В конкретных случаях может применяться одно или сразу несколько из перечисленных ограничений. Отметим при этом, что введение ограничений лишь на размер программ является недостаточным, поскольку не позволяет отделить длинные, но простые фрагменты от коротких, но запутанных.

В большинстве существующих пакетов и систем модули оформляются в виде программных единиц. В качестве языка представления модулей обычно используется язык загрузки или какой-либо язык программирования. Существуют пакеты, которые допускают использование текстовых модулей на различных языках программирования. Предметное обеспечение пакетов, использующих лишь модули типа «программная единица», организовано по принципу библиотеки программ.

При решении сложных задач, требующих большого количества модулей в различных сочетаниях, такой подход к организации предметного обеспечения может вызвать значительные технические трудности на этапе сборки программ. Например, весьма трудоемким для пользователя является выполнение таких работ, как редактирование текстов модулей (с использованием штатного редактора текстов), их трансляция, формирование библиотеки модулей для конкретного расчетного варианта и т. п. Для облегчения выполнения этих действий, а также в целях их частичной или полной автоматизации в ряде современных пакетов в качестве модулей допускается использование таких элементов, как фрагмент текста на языке программирования, схема счета задачи, объект данных, описание свойств (паспорт) модуля и др. Сказанное можно проиллюстрировать на примере ППП (САФРА) [14], ориентированного на широкий класс задач газовой динамики.

Составным элементом пакета САФРА является специализированный архив, предназначенный для хранения накапливаемого программного

фонда и формулируемой на языке заданий информации о структуре собираемых программ. Хранимые в архиве единицы называются модулями пакета. Каждый модуль характеризуется наличием у него архивного имени и возможностью независимого доступа. Архивное имя – важный объект пакета, так как после определенных автоматических преобразований задание на сборку сводится к перечислению в нужном порядке архивных имен модулей, из которых составляются программа и ее начальные данные. При этом в программных единицах указываются не архивные, а функциональные имена модулей, отражающие функциональный смысл того или иного модуля и определяющие место такого модуля в общей структуре программы. Этим достигается гибкость использования программного фонда и возможность широкого варьирования сочетаний модулей в программах. Каждое функциональное имя относится к некоторому классу модулей, имеющих сходное назначение. В языке заданий имеются средства указания соответствия между функциональными и архивными именами модулей для задания конкретных вариантов сборки.

В пакете САФРА допускаются следующие типы модулей: программная единица, вставка, данные, схема счета, версия, вариант, единица задания.

Модуль «программная единица» – это программа на языке программирования: подпрограмма или подпрограмма-функция. Такой модуль может также содержать и специальные управляющие предложения языка заданий, которые, например, могут указывать на то, что в данное место модуля нужно вставить некоторую последовательность предложений, которая в пакете описана в виде модуля типа «вставка».

Модуль «вставка» предназначен для оформления одного или нескольких предложений языка программирования или языка заданий, которые могут служить вставками, например, в программные единицы или в модули других типов. Обычно в виде вставок оформляются последова-

тельности операторов, имеющие относительно самостоятельное значение и определяющие те локальные решения, которые могут изменяться в различных программных комбинациях. Каждая вставка состоит из нескольких предложений, но не обязательно является программой, допускающей автономную трансляцию. Вставка может содержать ссылки на другие вставки.

Модуль «данные» служит для оформления числовой или иной информации, которая может быть использована в качестве данных при счете программы.

Модуль «схема счета» предназначен для запоминания перечня основных функциональных элементов, на которые расчленяется алгоритм. Этому перечню элементов соответствует набор программных единиц, участвующих в решении некоторого класса задач. Схема счета составляется в терминах функциональных имен в расчете на широкую свободу варьирования конкретных программ. При сборке каждому функциональному имени ставится в соответствие некоторое архивное имя модуля. Таким образом, схема счета аналогична принципиальной блок–схеме, составленной в содержательных наименованиях. В схему счета, хранимую в архиве, пользователи могут вносить уточнения, необходимые для конкретной реализации программы.

Модуль «версия» служит для описания конкретной версии программы, т. е. для установления соответствия между отдельными функциональными именами, указанными в схеме счета, и архивными именами соответствующих программных единиц.

Модуль "вариант" предназначен для описания конкретного счетного варианта программы. В нем должны быть конкретизированы все вставки в программу, исходные данные, необходимые ресурсы и т. д.

Разработка программы в рамках ППП (САФРА) начинается с составления общей схемы счета, которая представляет собой перечисление функциональных имен модулей, отражающих основные этапы решения за-

дачи. Схема счета может быть определена как явно, т. е. путем явного указания входящих в нее модулей, так и косвенно - посредством ссылки на некоторую разработанную ранее схему, дополненную функциональными именами вновь включаемых модулей.

Следующий шаг – разработка внутренней структуры и программных тел отдельных модулей. В процессе конструирования каждого модуля в нем могут быть выделены самостоятельные части, которые оформляются как модули более низкого уровня. Таким образом, осуществляется выделение и накопление модулей различных уровней.

Отладка программы может производиться сразу в нескольких направлениях, каждое из которых связано с отладкой группы модулей, определяющей некоторую версию программы. Такая организация отладки «сверху вниз» позволяет исключить этап автономной отладки модулей и проверить их работу в контексте всей программы.

ТЕМА 5. ПОДГОТОВКА И ВВОД ИСХОДНЫХ ДАННЫХ

5.1. Программы *ID-HCDz*

При решении проектных и исследовательских задач для дизелей различают прямую и обратную задачи. Прямая задача – определение геометрических, термодинамических, эффективных показателей дизеля при заданных параметрах окружающей среды (атмосферы, высоты над уровнем моря) и мощности. Обратная задача – определение термодинамических, эффективных показателей дизеля при заданных параметрах окружающей среды и его геометрических параметрах (диаметр и ход поршня).

Для решения прямой и обратной задач для дизеля была разработана система автоматизированного расчета (САР), построенная по модульному принципу как совокупность программных модулей. Система реализована в интегрированной среде *Mathcad* на ПЭВМ. Пакет прикладных программ разработан на основе работы [16].

На рис. 5.1 представлена подпрограмма ввода исходных данных для теплового расчета дизеля (прямая задача) *ID_TPDz_1.mcd*.

Имена переменных и их значения представлены в подпрограмме (рис. 5.1). В САР реализован один из возможных способов ввода исходных данных (ИД) – шаблон программы. Приведенные данные служат данными для тестирования отдельных программных модулей и САР в целом. (При расчете дизеля с другими исходными параметрами использованный способ ввода ИД позволяет оперативно изменять весь комплекс данных.)

Имена переменных математической модели (ММ) теплового расчета дизеля и идентификаторы в программах практически полностью соответствуют обозначениям работы [16], что существенно облегчает работу с программными единицами, текстом методической разработки и ММ.

Как можно видеть из текста программы (рис. 5.1, рис. 5.2), все вводимые значения переменных записаны в матрицу *MID_HCD*, а содержимое матрицы с помощью функции *WRITEPRN* записано в текстовый структурированный файл *MTR_MID_HCD_1.prn*.

ПОДПРОГРАММА

ввода исходных данных для теплового расчета дизеля

(прямая задача)

ID_TPDz_1.mcd

N_e := 60 - эффективная мощность дизеля, кВт;
 $n_{дв}$:= 1600 - число оборотов двигателя, об/мин;
 $i_{ц}$:= 4 - число цилиндров двигателя;
 i_{sd} := 1.1 - ход поршня относительный;
 α := 1.6 - коэффициент избытка воздуха;
 p_0 := 0.103 - давление начальное во впускном коллекторе двигателя, МПа;
 T_0 := 293 - температура начальная во впускном коллекторе двигателя, К;
 p_k := 0.18 - давление наддува, МПа;
 p_x := 0.003 - потеря давления в охладителе наддувочного воздуха (ОНВ), МПа;
 ΔT_x := 20 - понижение температуры воздуха в ОНВ, град;
 k := 1.4 - показатель адиабаты для воздуха;
 ξ_{int} := 0.95 - коэффициент гидравлических потерь на впуске для дизелей;
 p_{kr} := 1.15 - отношение давления после компрессора к давлению перед турбиной;
 ΔT := 10 - подогрев заряда, град;
 T_r := 900 - температура остаточных газов, К;
 ε := 14 - степень сжатия;
 p_z := 8 - максимальное давление сгорания, МПа;
 ξ_z := 0.8 - коэффициент;
 ξ_b := 0.9 - коэффициент;
 $\phi_{п}$:= 0.93 - коэффициент полноты диаграммы для дизеля;
 $\eta_{ка}$:= 0.72 - КПД адиабатический компрессора;
 ξ_{ex} := 1.2 - коэффициент;

Рис. 5.1. Подпрограмма ввода исходных данных *ID_TPDz_1*(начало)

$n_1 := 1.36$ - показатель политропы;

$\eta_M := 0.87$ - механический КПД;

$D := 1.04$ - диаметр цилиндра двигателя, дм;

$S := D \cdot i_{sd}$ - ход поршня, дм;

$V_h := \pi \cdot \frac{D^2}{4} (D \cdot i_{sd})$ - рабочий объем цилиндра двигателя, дм³;

Состав топлива:

$C := 0.87$ - массовая доля углерода в топливе;

$H := 0.121$ - массовая доля водорода в топливе;

$Se := 0.005$ - массовая доля серы в топливе;

$O := 0.004$ - массовая доля кислорода в топливе;

$W := 0$ - массовая доля воды в топливе.

Формируем матрицу исходных данных

$$\text{MID_HCD} := \begin{pmatrix} N_e & n_{\text{дв}} & i_{\text{ц}} & i_{\text{sd}} & \alpha & p_0 \\ T_0 & p_k & p_x & \Delta T_x & k & \xi_{\text{int}} \\ p_{kr} & \Delta T & T_r & \varepsilon & p_z & \xi_z \\ \xi_b & \phi_{\text{п}} & \eta_{\text{ка}} & \xi_{\text{ex}} & n_1 & \eta_M \\ 0 & 0 & 0 & C & H & Se \\ O & W & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{MID_HCD} = \begin{pmatrix} 60 & 1.6 \times 10^3 & 4 & 1.1 & 1.6 & 0.103 \\ 293 & 0.18 & 3 \times 10^{-3} & 20 & 1.4 & 0.95 \\ 1.15 & 10 & 900 & 14 & 8 & 0.8 \\ 0.9 & 0.93 & 0.72 & 1.2 & 1.36 & 0.87 \\ 0 & 0 & 0 & 0.87 & 0.121 & 5 \times 10^{-3} \\ 4 \times 10^{-3} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Записываем матрицу исходных данных в текстовой структурированный файл
WRITEPRN("MTR_MID_HCD_1.prn") := MID_HCD **Конец программы**

Рис. 5.1. Подпрограмма *ID_TPDz_1* (окончание)

Текстовый файл может быть использован в отчетном документе или для расчетов в других программных модулях (ПМ), если это необходимо. В этом случае данные текстового файла считываются в ПМ функцией *READPRN*.

На рис. 5.1 видно, что параметры D , S , V_h – диаметр поршня, ход поршня, рабочий объем цилиндра соответственно выключены из расчетов (черный квадратик в правом верхнем углу обозначения переменной). Эти параметры «включаются» при решении обратной задачи.

Программный модуль *ID_TPDz_2.mcd*, используемый для обратной задачи, во многом идентичен вышерассмотренному. На рис. 5.2 представлен текст подпрограммы ввода исходных данных теплового расчета дизеля при решении обратной задачи. Из текста ПМ видно, что имена переменных и идентификаторы сохранены, что облегчает работу с программами.

5. 2. Международная стандартная атмосфера

Известно, что состояние атмосферы не характеризуется большой определенностью [17]: и давление, и температура, и плотность воздуха колеблются в достаточно широких пределах, завися от ряда факторов, которые в настоящее время еще не поддаются учету. В то же время в практических задачах необходимо уметь сравнивать отдельные результаты испытаний самолетов, моторов, двигателей, проведенных в различных местах и в разное время. Не менее остра необходимость в унификации различных расчетов, связанных с двигателями, в том числе и дизелями. Все это приводит к необходимости создания каких-то условных характеристик атмосферы, которые, являясь международным стандартом, могли бы обеспечить сравнимость как результатов испытаний различного типа двигателей, так и результатов их расчетов.

ПОДПРОГРАММА

ввода исходных данных для теплового расчета дизеля
(обратная задача)

ID_TPDz_2.mcd

$N_e := 60$ - эффективная мощность дизеля, кВт;

$n_{дв} := 1600$ - число оборотов двигателя, об/мин;

$i_{ц} := 4$ - число цилиндров двигателя;

$i_{sd} := 1.1$ - ход поршня относительный;

$\alpha := 1.6$ - коэффициент избытка воздуха;

$p_0 := 0.103$ - давление начальное во впускном коллекторе двигателя, МПа;

$T_0 := 293$ - температура начальная во впускном коллекторе двигателя, К;

$p_k := 0.18$ - давление наддува, МПа;

$p_x := 0.003$ - потеря давления в охладителе наддувочного воздуха (ОНВ),
МПа;

$\Delta T_x := 20$ - понижение температуры воздуха в ОНВ, град;

$k := 1.4$ - показатель адиабаты для воздуха;

$\xi_{int} := 0.95$ - коэффициент гидравлических потерь на впуске для дизелей;

$p_{kr} := 1.15$ - отношение давления после компрессора к давлению перед
турбиной;

$\Delta T := 10$ - подогрев заряда, град;

$T_r := 900$ - температура остаточных газов, К;

$\varepsilon := 14$ - степень сжатия;

$p_z := 8$ - максимальное давление сгорания, МПа;

$\xi_z := 0.8$ - коэффициент;

$\xi_b := 0.9$ - коэффициент;

$\phi_{п} := 0.93$ - коэффициент полноты диаграммы для дизеля;

$\eta_{ка} := 0.72$ - КПД адиабатический компрессора;

$\xi_{ex} := 1.2$ - коэффициент;

Рис. 5.2. Подпрограмма *ID_TPDz_2.mcd* (обратная задача) (начало)

$n_1 := 1.36$ - показатель политропы;

$\eta_M := 0.87$ - механический КПД;

$D := 1.04$ - диаметр цилиндра двигателя, дм;

$S := D \cdot i_{sd}$ - ход поршня, дм;

$V_h := \pi \cdot \frac{D^2}{4} (D \cdot i_{sd})$ - рабочий объем цилиндра двигателя, дм³;

Состав топлива:

$C := 0.87$ - массовая доля углерода в топливе;

$H := 0.121$ - массовая доля водорода в топливе;

$Se := 0.005$ - массовая доля серы в топливе;

$O := 0.004$ - массовая доля кислорода в топливе;

$W := 0$ - массовая доля воды в топливе.

Формируем матрицу исходных данных

$$\text{MID_HCD} := \begin{pmatrix} \text{"Ne"} & n_{\text{дв}} & i_{\text{ц}} & i_{\text{sd}} & \alpha & p_0 \\ T_0 & p_k & p_x & \Delta T_x & k & \xi_{\text{int}} \\ p_{kr} & \Delta T & T_r & \varepsilon & p_z & \xi_z \\ \xi_b & \phi_{\text{п}} & \eta_{\text{ка}} & \xi_{\text{ex}} & n_1 & \eta_M \\ D & 0 & 0 & C & H & Se \\ O & W & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{MID_HCD} = \begin{pmatrix} \text{"Ne"} & 1.6 \times 10^3 & 4 & 1.1 & 1.6 & 0.103 \\ 293 & 0.18 & 3 \times 10^{-3} & 20 & 1.4 & 0.95 \\ 1.15 & 10 & 900 & 14 & 8 & 0.8 \\ 0.9 & 0.93 & 0.72 & 1.2 & 1.36 & 0.87 \\ 1.04 & 0 & 0 & 0.87 & 0.121 & 5 \times 10^{-3} \\ 4 \times 10^{-3} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Записываем матрицу исходных данных в текстовый структурированный файл
`WRITEPRN("MTR_MID_HCD_2.prn") := MID_HCD` **Конец программы**

Рис. 5.2. Подпрограмма *ID_TPDz_2.mcd* (окончание)

Такие характеристики были предложены в форме Международной Стандартной Атмосферы (МСА). МСА в наибольшей степени подходит к условиям средних северных широт земного шара (35°N — 60°N), соответствующая, таким образом, атмосферным данным наиболее населенных областей Земли.

При разработке МСА были приняты определенные законы распределения температуры по высоте, полученные эмпирическим путем в результате исследования атмосферы шарами-зондами. Так, в тропосфере был принят линейный закон падения температуры по высоте:

$$t_H = 15 - 0,0065H,$$

где H — высота в м; t_H — температура в $^{\circ}\text{C}$.

В стратосфере, нижняя граница которой была принята равной 11 км, принят закон постоянства температуры ($t = -56,5^{\circ}\text{C}$).

Принимая $B_0 = 760$ мм рт. ст., $T_0 = 288^{\circ}$ абс. и $\rho_0 = 1/8$ кг м⁻⁴ сек², в работах [17, 18] для давления и плотности, получены следующие формулы для расчета Международной Стандартной Атмосферы для высот до 11 км (тропосфера):

$$\frac{P}{P_0} := \left(1 - \frac{H}{44300}\right)^{5.256};$$

$$\frac{\rho}{\rho_0} := \left(1 - \frac{H}{44300}\right)^{4.256}$$

Приведенные формулы дают возможность определить параметры окружающей среды (атмосферы) при изменении высоты над уровнем моря (H). Это необходимо, например, при эксплуатации, испытаниях или при проведении численного эксперимента для дизелей в условиях высокогорья, когда высотное положение двигателя может изменяться в процессе движе-

ния транспортного средства за короткое время на большую величину. При этом могут существенно изменяться мощностные, экономические и экологические показатели двигателя.

Для проведения теплового расчета дизеля при меняющихся атмосферных условиях и высоты над уровнем моря была разработана подпрограмма, являющаяся программным модулем CAP, *Pp_MATM.mcd*.

На рис. 5.3 представлен текст подпрограммы.

$$t_H(H) := 15 - 0.0065 \cdot H$$

где H – высота в м; t_H – температура в °С.

Формулы для расчета Международной Стандартной Атмосферы для высот до 11 км (тропосфера).

Начальные условия для построения графиков:

$$p_0 := 1 \quad \text{– давление; } \rho_0 := 1 \quad \text{– плотность;}$$

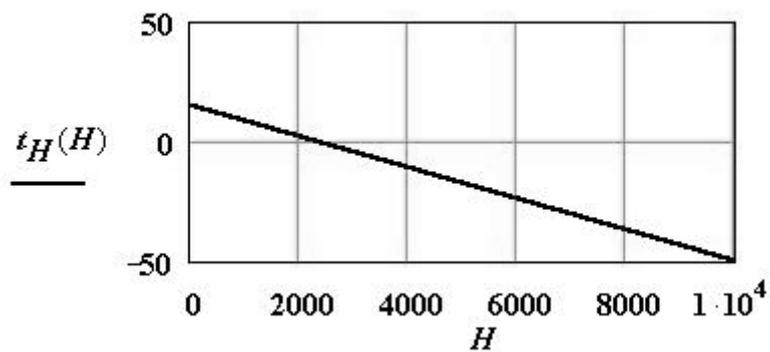
$$H := 0,100..11000 \quad \text{– изменение высоты, м.}$$

Относительное изменение давления в зависимости от высоты.

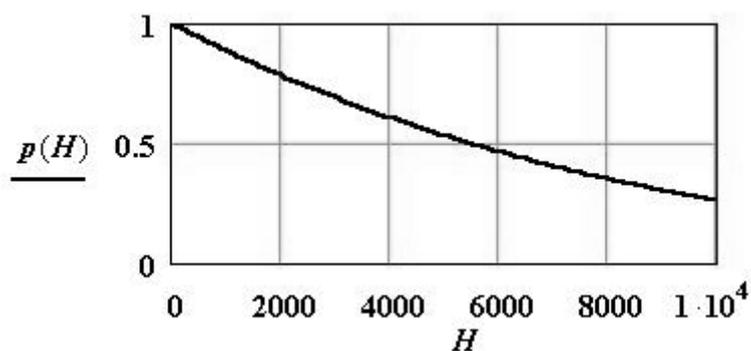
$$p(H) := p_0 \cdot \left(1 - \frac{H}{44300}\right)^{5.256}$$

Относительное изменение плотности в зависимости от высоты.

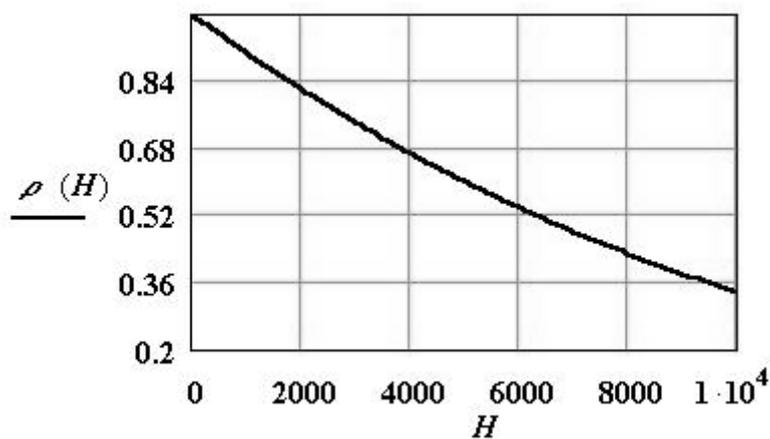
$$\rho(H) := \rho_0 \cdot \left(1 - \frac{H}{44300}\right)^{4.256}$$



Изменение температуры по высоте для стандартной атмосферы



Изменение относительного давления по высоте для стандартной атмосферы



Изменение относительной плотности по высоте для стандартной атмосферы

Рис. 5.3. Подпрограмма расчета параметров МСА

ТЕМА 6. ПРОГРАММНЫЕ МОДУЛИ ТОПЛИВА И ТЕПЛОВОГО РАСЧЕТА

В данном разделе приведены тексты программных модулей, предназначенных для расчета параметров топлива, продуктов сгорания и теплового расчета цикла дизеля.

Подпрограмма *Pp_FUEL.mcd* (рис. 6.1) включает пять функций пользователя:

- функция $Hu(V_f)$ – расчет низшей теплоты сгорания топлива;
- $L_0(V_f)$ – расчет стехиометрического коэффициента для топлива;
- $M_0(V_f)$ – расчет количества продуктов сгорания на 1 кг топлива при $\alpha=1$;
- $U_{возд}(t)$, $U_{псз}(t)$ – расчет внутренней энергии воздуха и продуктов сгорания соответственно в зависимости от температуры.

Все указанные функции записаны в параметрическом виде, где формальными параметрами являются вектор состава топлива V_f и температура t °С.

Подпрограмма *Pp_HCD_1.mcd* (рис. 6.2) выполняет тепловой расчет дизеля (прямая задача $N_e = \text{const}$).

Входными параметрами этой подпрограммы являются данные файлов *ID_TPDz_1.mcd* и *Pp_FUEL.mcd.*, связанных функцией *Reference* с *Pp_HCD_1.mcd*.

Выходные данные, оформленные в виде матрицы 1X14

$$HD(\alpha p_k p_0 T_0) = (\alpha p_k p_0 T_0 \eta_e g_e p_e T_z h_u \eta_i p_i D S N_e),$$

содержащей результаты расчетов, включают следующие переменные:

- α - коэффициент избытка воздуха;
- p_k - давление наддува;
- p_0 - начальное давление (атмосферное давление);

- T_0 - начальная температура;
- η_e - эффективный КПД дизеля;
- g_e - эффективный расход топлива;
- T_z - максимальная температура цикла;
- h_u - низшая теплота сгорания топлива;
- η_i - индикаторный КПД цикла дизеля;
- p_i - индикаторное давление;
- D - диаметр поршня цилиндра дизеля;
- S - ход поршня;
- N_e – эффективная мощность дизеля.

На рис.6.3 представлена подпрограмма *Pp_HCD_2.mcd*, предназначенная для теплового расчета дизеля при решении обратной задачи проектирования ($N_e=const$, $D=const$, $S=const$). Информационные связи модуля *Pp_HCD_2.mcd* и структура матрицы выходных данных во многом аналогичны, модулю *Pp_HCD_1.mcd*.

Следует иметь ввиду, что в соответствии с решаемой задачей исходные данные для модуля *Pp_HCD_2.mcd* несколько отличаются, что достаточно хорошо видно из текста программ модулей *ID_TPDz_1.mcd* и *ID_TPDz_1.mcd*, представленных на рис. 5.1 – рис. 5.4.

$$H_u(V_f) := \begin{cases} h \leftarrow V_{f_0} \cdot 33910 \\ h \leftarrow h + V_{f_1} \cdot 125600 \\ h \leftarrow h + (V_{f_2} - V_{f_3}) \cdot 10890 \\ h \leftarrow h + (9 \cdot V_{f_1} - V_{f_4}) \\ h \end{cases} \quad \begin{array}{l} \text{Низшая теплота сгорания} \\ \text{топлива (рассчитывается} \\ \text{по формуле Д.И. Менде-} \\ \text{леева)} \end{array}$$

Рис. 6.1. Подпрограмма *Pp_FUEL.mcd* (начало)

$$L_{\theta}(V_f) := \left| \begin{array}{l} s \leftarrow \frac{V_{f_0}}{12} \\ s \leftarrow s + \frac{V_{f_1}}{4} \\ s \leftarrow s + \frac{(V_{f_2} - V_{f_3})}{32} \\ s \leftarrow \frac{s}{0.21} \\ s \end{array} \right.$$

Теоретически
необходимое количество
воздуха для сгорания 1 кг
топлива

$$M_{\theta}(V_f) := \left| \begin{array}{l} s \leftarrow \frac{V_{f_0}}{12} \\ s \leftarrow s + \frac{V_{f_1}}{2} \\ s \leftarrow s + 0.79 \cdot L_{\theta}(V_f) \\ s \end{array} \right.$$

Количество продуктов
сгорания на 1 кг топлива
при $\alpha=1$

$$U_{\text{гозд}}(t) := 0.0017387 \cdot t^2 + 21.929 \cdot t - 289.14$$

$$U_{\text{ncz}}(t) := 0.0024012 \cdot t^2 + 24.418 \cdot t - 415.43$$

Тест подпрограммы расчета характеристик топлива и продуктов сгорания

$$\mathbf{V_T} := (0.87 \quad 0.121 \quad 0.005 \quad 0.004 \quad 0)$$

$$H_u(\mathbf{V_T}^T) = 4.471 \times 10^4 \quad L_0(\mathbf{V_T}^T) = 0.489 \quad M_0(\mathbf{V_T}^T) = 0.52$$

Рис. 6.1. Подпрограмма *Pp_FUEL.mcd* (окончание)

⊕ Reference:E:\Тепловой расчет дизеля высокогорье\ID_TPDz_1.mcd

⊕ Reference:E:\Тепловой расчет дизеля высокогорье\Pr_FUEL.mcd

$$V_f := (C \ H \ Se \ O \ W) \quad V_f := V_f^T$$

$$\begin{aligned}
 HD(\alpha, p_k, p_0, T_0) := & \left\{ \begin{aligned}
 & h_u \leftarrow H_u(V_f) \\
 & l_0 \leftarrow L_0(V_f) \\
 & m_0 \leftarrow M_0(V_f) \\
 & M_1 \leftarrow \alpha \cdot L_0(V_f) \\
 & M_2 \leftarrow M_0(V_f) + (\alpha - 1) \cdot L_0(V_f) \\
 & r_0 \leftarrow \frac{M_0(V_f)}{[M_0(V_f) + (\alpha - 1) \cdot L_0(V_f)]} \\
 & r_\alpha \leftarrow \frac{[(\alpha - 1) \cdot L_0(V_f)]}{M_2} \\
 & \mu_0 \leftarrow \frac{M_2}{M_1} \\
 & p'_k \leftarrow p_k + p_x \\
 & T_k \leftarrow T_0 \cdot \left[1 + \frac{\left(\frac{p'_k}{p_0}\right)^{\frac{k-1}{k}} - 1}{\eta_{ka}} \right] - \Delta T_x \\
 & p_a \leftarrow \xi_{int} \cdot p_k \\
 & p_r \leftarrow \xi_{ex} \cdot \frac{p_k}{p_{kr}} \\
 & \gamma \leftarrow \left(\frac{T_k + \Delta T}{T_r} \right) \cdot \frac{p_r}{\varepsilon \cdot p_a - p_r}
 \end{aligned} \right.
 \end{aligned}$$

Рис. 6.2. Подпрограмма Pr_HCDz_1.mcd прямая задача (начало)

$$T_a \leftarrow \frac{T_k + \Delta T + \gamma \cdot T_r}{1 + \gamma}$$

$$\eta_v \leftarrow \frac{\varepsilon}{\varepsilon - 1} \cdot \frac{P_a}{P_k} \cdot \frac{T_k}{(T_k + \Delta T)} \cdot \left(1 - \frac{P_r}{\varepsilon \cdot P_a} \right)$$

$$P_c \leftarrow P_a \cdot \varepsilon^{n_I}$$

$$T_c \leftarrow T_a \cdot \varepsilon^{n_I - 1}$$

$$\lambda \leftarrow \frac{P_z}{P_c}$$

$$\mu \leftarrow \frac{\mu_0 + \gamma}{1 + \gamma}$$

$$q_1 \leftarrow \frac{1 + \gamma \cdot r_\alpha}{1 + \gamma}$$

$$q_2 \leftarrow \frac{\gamma \cdot r_0}{1 + \gamma}$$

$$U_{лев} \leftarrow \xi_z \cdot \frac{h_u}{(1 + \gamma) \cdot M_I} + 0 \dots$$

$$+ q_1 \cdot U_{возд}(T_c - 273) \dots$$

$$+ 0 + q_2 \cdot U_{ncz}(T_c - 273) + 8.314 \cdot \lambda \cdot T_c$$

$$t \leftarrow 1700$$

$$t_z \leftarrow \text{root} \left[\begin{array}{l} U_{лев} \dots \\ + 0 - \mu \cdot \left[\begin{array}{l} r_\alpha \cdot U_{возд}(t) \dots \\ + 0 + r_0 \cdot U_{ncz}(t) \dots \\ + 0 + 8.314 \cdot (t + 273) \end{array} \right] \end{array} \right], t$$

$$T_z \leftarrow t_z + 273$$

$$\rho \leftarrow \mu \cdot \frac{T_z}{\lambda \cdot T_c}$$

Рис. 6.2. Подпрограмма *Pp_HCDz_1.mcd* (продолжение)

$$\delta \leftarrow \frac{\varepsilon}{\rho}$$

$$U'' \leftarrow (\xi_b - \xi_z) \cdot \frac{h_u}{(\mu_0 + \gamma) \cdot M_1}$$

$$n \leftarrow 1.15$$

$$n_2 \leftarrow \text{root} \left[\begin{array}{l} U'' \dots \\ + 0 - \left[\begin{array}{l} \frac{(T_z - T_z \cdot \delta^{1-n})}{n-1} \dots \\ + 0 - r_\alpha \cdot \left(\begin{array}{l} U_{6030}(t_z) \dots \\ + 0 - U_{6030} \left(\begin{array}{l} T_z \cdot \delta^{1-n} \dots \\ + 0 - 273 \end{array} \right) \end{array} \right) \dots \\ + 0 - r_0 \cdot \left(U_{nc2}(t_z) - U_{nc2} \left(\begin{array}{l} T_z \cdot \delta^{1-n} - 273 \end{array} \right) \right) \end{array} \right] \end{array} \right], n$$

$$p_b \leftarrow p_z \cdot \delta^{-n_2}$$

$$T_b \leftarrow T_z \cdot \delta^{1-n_2}$$

$$p_{ip} \leftarrow \frac{p_c}{\varepsilon - 1} \cdot \left[\begin{array}{l} \lambda \cdot (\rho - 1) \dots \\ + 0 + \lambda \cdot \frac{\rho}{n_2 - 1} \cdot \left(1 - \frac{T_b}{T_z} \right) \dots \\ + 0 - \frac{1}{n_1 - 1} \cdot \left(1 - \frac{T_a}{T_c} \right) \end{array} \right]$$

$$p_i \leftarrow \phi_n \cdot p_{ip}$$

$$\eta_i \leftarrow 8.314 \cdot \frac{p_i \cdot M_1 \cdot T_k}{h_u \cdot \eta_v \cdot p_k}$$

$$g_i \leftarrow \frac{3600000}{h_u \cdot \eta_i}$$

$$\eta_e \leftarrow \eta_i \cdot \eta_M$$

$$p_e \leftarrow p_i \cdot \eta_M$$

Рис. 6.2. Подпрограмма Pp_HCDz_1.mcd (продолжение)

$$\begin{aligned}
 g_e &\leftarrow \frac{3600000}{h_u \cdot \eta_e} \\
 V_h &\leftarrow 120 \cdot \frac{N_e}{p_e \cdot n_{\text{об}} \cdot i_u} \\
 D &\leftarrow \sqrt[3]{\frac{V_h}{0.785 \cdot i_{sd}}} \\
 S &\leftarrow D \cdot i_{sd} \\
 &\left(\alpha \ p_k \ p_0 \ T_0 \ \eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \right)
 \end{aligned}$$

Конец программы теплового расчета дизеля

Тест программы теплового расчета дизеля

$$\left(\alpha \ p_k \ p_0 \ T_0 \ \eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ N_e \right)^{\blacksquare}$$

$$HD[2, 0.11, (p_0), (T_0)] =$$

	0	1	2	3	4	5
0	2	0.11	0.1	293	0.39	205.99

$$HD[2, 0.11, (p_0 \cdot 0.85), (T_0 - 10)] =$$

	1	2	3	4	5
0	0.11	0.088	283	0.391	206

Рис. 6.2. Подпрограмма Pp_HCDz_1.mcd (окончание)

ПОДПРОГРАММА

теплового расчета дизеля-2

$N=var$,

☞ Reference:Е:\Тепловой расчет дизеля высокогорья\ID_TPDz_2.mcd

☞ Reference:Е:\Тепловой расчет дизеля высокогорья\Pr_FUEL.mcd

$$NEVAR(\alpha, p_k, p_0, T_0) := \left[\begin{array}{l} h_u \leftarrow H_u(V_f) \\ l_0 \leftarrow L_0(V_f) \\ m_0 \leftarrow M_0(V_f) \\ M_1 \leftarrow \alpha \cdot L_0(V_f) \\ M_2 \leftarrow M_0(V_f) + (\alpha - 1) \cdot L_0(V_f) \\ r_0 \leftarrow \frac{M_0(V_f)}{[M_0(V_f) + (\alpha - 1) \cdot L_0(V_f)]} \\ r_\alpha \leftarrow \frac{[(\alpha - 1) \cdot L_0(V_f)]}{M_2} \\ \mu_0 \leftarrow \frac{M_2}{M_1} \\ p'_k \leftarrow p_k + p_x \\ T_k \leftarrow T_0 \cdot \left[1 + \frac{\left(\frac{p'_k}{p_0}\right)^{\frac{k-1}{k}} - 1}{\eta_{ka}} \right] - \Delta T_x \\ p_a \leftarrow \xi_{int} \cdot p_k \\ p_r \leftarrow \xi_{ex} \cdot \frac{p_k}{p_{kr}} \end{array} \right.$$

Рис. 6.3. Подпрограмма Pr_HCD_2.mcd (начало)

$$\gamma \leftarrow \left(\frac{T_k + \Delta T}{T_r} \right) \cdot \frac{p_r}{\varepsilon \cdot p_a - p_r}$$

$$T_a \leftarrow \frac{T_k + \Delta T + \gamma \cdot T_r}{1 + \gamma}$$

$$\eta_v \leftarrow \frac{\varepsilon}{\varepsilon - 1} \cdot \frac{p_a}{p_k} \cdot \frac{T_k}{(T_k + \Delta T)} \cdot \left(1 - \frac{p_r}{\varepsilon \cdot p_a} \right)$$

$$p_c \leftarrow p_a \cdot \varepsilon^{n_1}$$

$$T_c \leftarrow T_a \cdot \varepsilon^{n_1 - 1}$$

$$\lambda \leftarrow \frac{p_z}{p_c}$$

$$\mu \leftarrow \frac{\mu_0 + \gamma}{1 + \gamma}$$

$$q_1 \leftarrow \frac{1 + \gamma \cdot r_\alpha}{1 + \gamma}$$

$$q_2 \leftarrow \frac{\gamma \cdot r_0}{1 + \gamma}$$

$$U_{\text{лев}} \leftarrow \xi_z \cdot \frac{h_u}{(1 + \gamma) \cdot M_1} + 0 \dots$$

$$+ q_1 \cdot U_{\text{возд}}(T_c - 273) \dots$$

$$+ 0 + q_2 \cdot U_{\text{пер}}(T_c - 273) \dots$$

$$+ 0 + 8.314 \cdot \lambda \cdot T_c$$

$$t \leftarrow 1700$$

Рис. 6.3. Подпрограмма *Pp_HCD_2.mcd* (продолжение)

$$\begin{aligned}
& t_z \leftarrow \text{root} \left[\begin{array}{l} U_{\text{лев}} \dots \\ + 0 - \mu \cdot \left[\begin{array}{l} r_\alpha \cdot U_{\text{возд}}(t) \dots \\ + 0 + r_0 \cdot U_{\text{псг}}(t) \dots \\ + 0 + 8.314 \cdot (t + 273) \end{array} \right] \end{array} \right], t \\
& T_z \leftarrow t_z + 273 \\
& \rho \leftarrow \mu \cdot \frac{T_z}{\lambda \cdot T_c} \\
& \delta \leftarrow \frac{\varepsilon}{\rho} \\
& U'' \leftarrow (\xi_b - \xi_z) \cdot \frac{h_u}{(\mu_0 + \gamma) \cdot M_1} \\
& n \leftarrow 1.15 \\
& n_2 \leftarrow \text{root} \left[\begin{array}{l} U'' \dots \\ + 0 - \left[\begin{array}{l} 8.314 \cdot \frac{(T_z - T_z \cdot \delta^{1-n})}{n-1} \\ + r_\alpha \cdot \left(\begin{array}{l} U_{\text{возд}}(t_z) \dots \\ + 0 - U_{\text{возд}}(T_z \cdot \delta^{1-n} - 273) \end{array} \right) \dots \\ + 0 - r_0 \cdot (U_{\text{псг}}(t_z) - U_{\text{псг}}(T_z \cdot \delta^{1-n} - 273)) \end{array} \right] - 0 \dots \end{array} \right], n \\
& p_b \leftarrow p_z \cdot \delta^{-n_2} \\
& T_b \leftarrow T_z \cdot \delta^{1-n_2} \\
& p_{ip} \leftarrow \frac{p_c}{\varepsilon - 1} \cdot \left[\begin{array}{l} \lambda \cdot (\rho - 1) \dots \\ + 0 + \lambda \cdot \frac{\rho}{n_2 - 1} \cdot \left(1 - \frac{T_b}{T_z} \right) \dots \\ + 0 - \frac{1}{n_1 - 1} \cdot \left(1 - \frac{T_a}{T_c} \right) \end{array} \right]
\end{aligned}$$

Рис. 6.3. Подпрограмма *Pp_HCD_2.mcd* (продолжение)

$$\begin{aligned}
p_i &\leftarrow \phi_{\Pi} \cdot p_{ip} \\
\eta_i &\leftarrow 8.314 \cdot \frac{p_i \cdot M_1 \cdot T_k}{h_u \cdot \eta_v \cdot p_k} \\
g_i &\leftarrow \frac{3600000}{h_u \cdot \eta_i} \\
\eta_e &\leftarrow \eta_i \cdot \eta_M \\
p_e &\leftarrow p_i \cdot \eta_M \\
g_e &\leftarrow \frac{3600000}{h_u \cdot \eta_e} \\
V_h &\leftarrow \pi \cdot \frac{D^2}{4} (D \cdot i_{sd}) \\
N_e &\leftarrow V_h \cdot \frac{(p_e \cdot n_{ДВ} \cdot i_u)}{120} \\
S &\leftarrow D \cdot i_{sd} \\
& \left(\alpha \quad p_k \quad p_0 \quad T_0 \quad \eta_e \quad g_e \quad p_e \quad T_z \quad h_u \quad \eta_i \quad p_i \quad D \quad S \quad N_e \quad V_h \right)
\end{aligned}$$

Тест теплового расчета

$$\left(\alpha \quad p_k \quad p_0 \quad T_0 \quad \eta_e \quad g_e \quad p_e \quad T_z \quad h_u \quad \eta_i \quad p_i \quad D \quad S \quad N_e \quad V_h \right)^{\blacksquare}$$

$$NEVAR[1.8, 0.11, (p_0), (T_0)] = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 1.8 & 0.1 & 0.1 & 293 & 0.1 & 676.5 & 0.3 & 2 \cdot 10 \\ \hline \end{array}$$

$$NEVAR[1.5, 0.16, (p_0), (T_0)] = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 1.5 & 0.2 & 0.1 & 293 & 0.3 & 315.4 & 0.8 & 2.2 \cdot 10 \\ \hline \end{array}$$

Конец программы теплового расчета дизеля

Рис. 6.3. Подпрограмма Pp_HCD_2.mcd (окончание)

ТЕМА 7. БЛОК ОКРУЖАЮЩЕЙ СРЕДЫ

7. 1. Блок атмосферного давления

Блок окружающей среды включает четыре программных модуля, решающих прямую и обратную задачи при исследовании теплового цикла дизеля при изменении атмосферных давления и температуры.

Подпрограмма *Pp_P0_1.mcd* (рис. 7.1) функцией *Reference* связана с подпрограммой *Pp_HCD_1.mcd*, что дает возможность использовать универсальный программный модуль решения прямой задачи при изменении начального давления атмосферы. Функция пользователя *MoutP0* в цикле создает матрицу результатов расчетов формата

$$MoutP0 := (\eta_e, g_e, p_e, T_z, h_u, \eta_i, p_i, D, S, V_h, \alpha, N_e, p_0),$$

где: α – коэффициент избытка воздуха; p_k – давление наддува; p_0 – начальное давление (атмосферное давление); T_0 – начальная температура; η_e – эффективный КПД дизеля; g_e – эффективный расход топлива; T_z – максимальная температура цикла; h_u – низшая теплота сгорания топлива; η_i – индикаторный КПД цикла дизеля; p_i – индикаторное давление; D – диаметр поршня цилиндра дизеля; S – ход поршня; N_e – эффективная мощность дизеля.

Подпрограмма *Pp_P0_1.mcd* формирует ряд векторов выходных данных, а затем производит интерполяцию кубическим сплайном векторов по величине начального атмосферного давления.

Подпрограмма *R_P0_1.mcd* (рис. 7.2) с помощью функции *READPRN* считывает данные текстовых структурированных файлов *MTR_HCDz.prn* и *MTR_fuel.prn*. Подпрограмма *R_P0_1.mcd* функцией *Reference* связана с подпрограммами *Pp_fuile.mcd* и *Pp_P0_1.mcd*. Результатом работы программы *R_P0_1.mcd* является расчет параметров цикла дизеля (прямая задача) при разном начальном давлении и графики зависимости параметров дизеля в функции от давления атмосферы.

Подпрограммы *Pp_P0_NVR_1.mcd* и *R_P0_NVR_1.mcd* (рис. 7.3 – рис. 7.4) структурно и информационно организованы аналогично программным модулям *Pp_P0_1.mcd* и *R_P0_1.mcd* с учетом того, что они решают обратную задачу.

7. 2. Блок начальной температуры

Подпрограмма *Pp_T0_1.mcd* (рис. 7.5) функцией *Reference* связана с подпрограммой *Pp_HCD_1.mcd*, что дает возможность использовать универсальный программный модуль решения прямой задачи при изменении начальной температуры атмосферы. Функция пользователя *MoutT0* в цикле создает матрицу результатов расчетов формата

$$MoutT0 := \eta_e g_e p_e T_z h_u \eta_i p_i D S V_h \alpha N_e p_0),$$

где: α – коэффициент избытка воздуха; p_k – давление наддува; p_0 – начальное давление (атмосферное давление); T_0 – начальная температура; η_e – эффективный КПД дизеля; g_e – эффективный расход топлива; T_z – максимальная температура цикла; h_u – низшая теплота сгорания топлива; η_i – индикаторный КПД цикла дизеля; p_i – индикаторное давление; D – диаметр поршня цилиндра дизеля; S – ход поршня; N_e – эффективная мощность дизеля.

Подпрограмма *Pp_T0_1.mcd* формирует ряд векторов выходных данных, а затем производит интерполяцию кубическим сплайном векторов по величине начальной температуры.

Подпрограмма *R_T0_1.mcd* (рис. 7.6) с помощью функции *READPRN* считывает данные текстовых структурированных файлов *MTR_HCDz.prn* и *MTR_fuel.prn*. Подпрограмма *R_T0_1.mcd* функцией *Reference* связана с подпрограммами *Pp_fuile.mcd* и

ПОДПРОГРАММА

теплового расчета дизеля $N=const$ при изменении начального давления воздуха

Reference:\ Pp_HCD_1.mcd

$HCDPO(p_0) := HD(\alpha, p_k, p_0, T_0)$

$MoutPO := \left\{ \begin{array}{l} p_0 \leftarrow 0.05 \\ Vout \leftarrow HCDPO(p_0) \\ \text{for } \varepsilon \in 1..11 \\ \quad Vout \leftarrow stack(Vout, HCDPO(p_0 + \varepsilon \cdot 0.005)) \\ Vout \end{array} \right.$

Формат матрицы выходных данных

$(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \ N_e \ p_0)$

Формирование векторов выходных параметров:

КПД эффективный $\eta E := MoutPO \langle 0 \rangle$;

начальное давление воздуха $PO := MoutPO \langle 12 \rangle$;

удельный эффективный расход топлива $gE := MoutPO \langle 1 \rangle$;

температура $Tz \ TZ := MoutPO \langle 3 \rangle$;

диаметр поршня $D \ DP := MoutPO \langle 7 \rangle$;

эффективная мощность $Ne \ NE := MoutPO \langle 11 \rangle$.

$\eta E(p_0) := interp(lspline(PO, \eta E), PO, \eta E, p_0)$

$gE(p_0) := interp(lspline(PO, gE), PO, gE, p_0)$

$Tz(p_0) := interp(lspline(PO, TZ), PO, TZ, p_0)$

$Dn(p_0) := interp(lspline(PO, DP), PO, DP, p_0)$

$NE(p_0) := interp(lspline(PO, NE), PO, NE, p_0)$

Рис. 7.1. Подпрограмма Pp_PO_1.mcd

Считывание данных матриц

$M_1 := \text{READPRN}(\text{"C:\Мои документы\TPDz\MTR_HCDz.prn"})$

$V_f := \text{READPRN}(\text{"C:\Мои документы\TPDz\MTR_fuel.prn"})$

⇒ *Reference:C:\Мои документы\TPDz\Pr_fuile.mcd*

⇒ *Reference:C:\Мои документы\TPDz\Pr_HCD_PO.mcd*

$MoutPO =$

	0	1	2	3	4	5	6
0	0.379	212.18	0.632	$2.407 \cdot 10^3$	$4.471 \cdot 10^4$	0.436	0.726
1	0.38	211.871	0.655	$2.372 \cdot 10^3$	$4.471 \cdot 10^4$	0.437	0.753
2	0.38	211.622	0.677	$2.341 \cdot 10^3$	$4.471 \cdot 10^4$	0.437	0.778
3	0.381	211.421	0.698	$2.313 \cdot 10^3$	$4.471 \cdot 10^4$	0.438	0.802
4	0.381	211.26	0.718	$2.287 \cdot 10^3$	$4.471 \cdot 10^4$	0.438	0.825
5	0.381	211.131	0.737	$2.264 \cdot 10^3$	$4.471 \cdot 10^4$	0.438	0.847
6	0.382	211.03	0.755	$2.243 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.868
7	0.382	210.952	0.773	$2.223 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.888
8	0.382	210.894	0.79	$2.205 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.908
9	0.382	210.853	0.807	$2.187 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.927
10	0.382	210.827	0.823	$2.171 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.946
11	0.382	210.813	0.838	$2.156 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.963

Формат матрицы

$(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \ N_e \ p_0)$;

результат теплового расчета дизеля $MTR_HCD_PO := MoutPO$

Запись результатов расчета в структурированный файл:

$WRITEPRN(\text{"C:\Мои документы\TPDz\MTR_HCD_PO.prn"}) := MTR_HCD_PO$

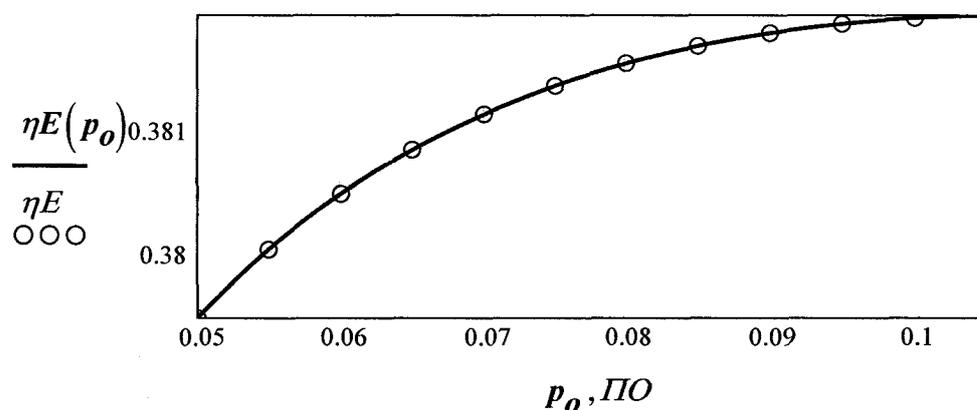
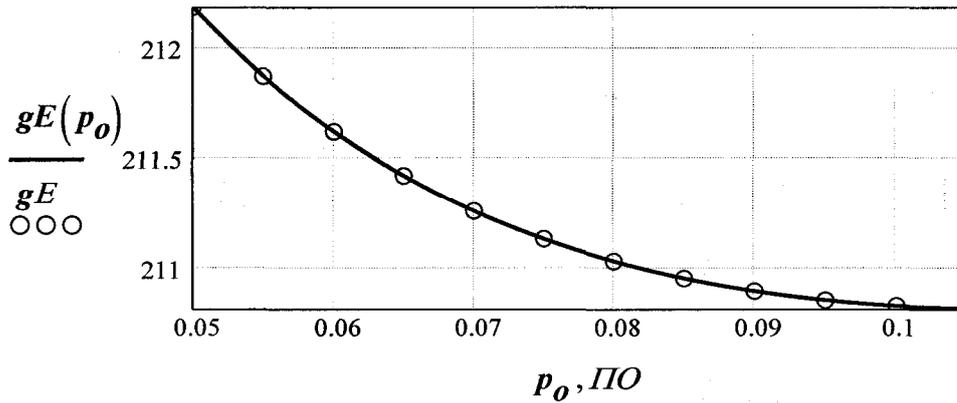
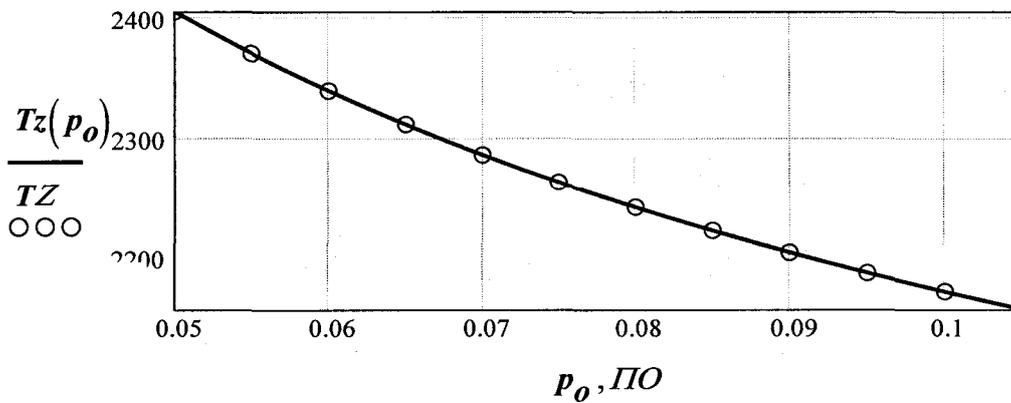


Рис. Изменение эффективного КПД в зависимости от начального давления воздуха

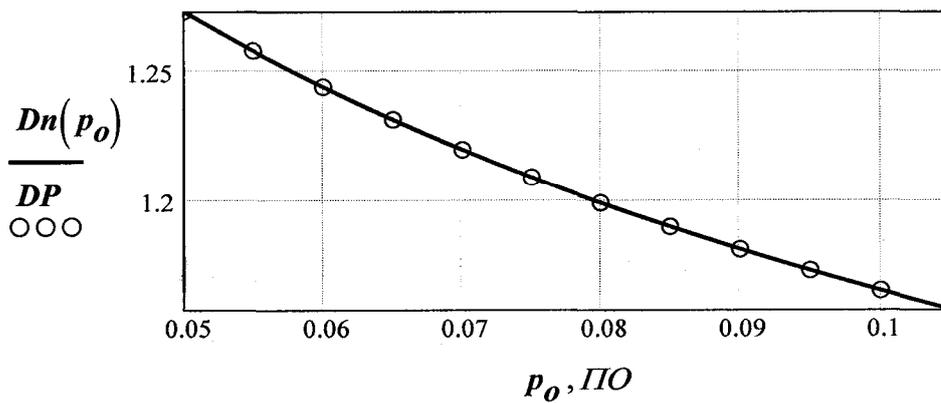
Рис. 7.2. Программа $R_PO_1.mcd$ (начало)



Изменение эффективного расхода топлива в зависимости от начального давления воздуха

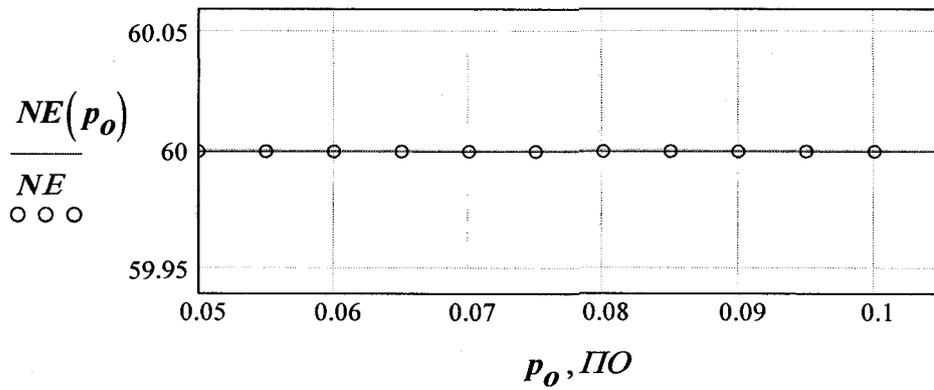


Изменение максимальной температуры в зависимости от начального давления воздуха

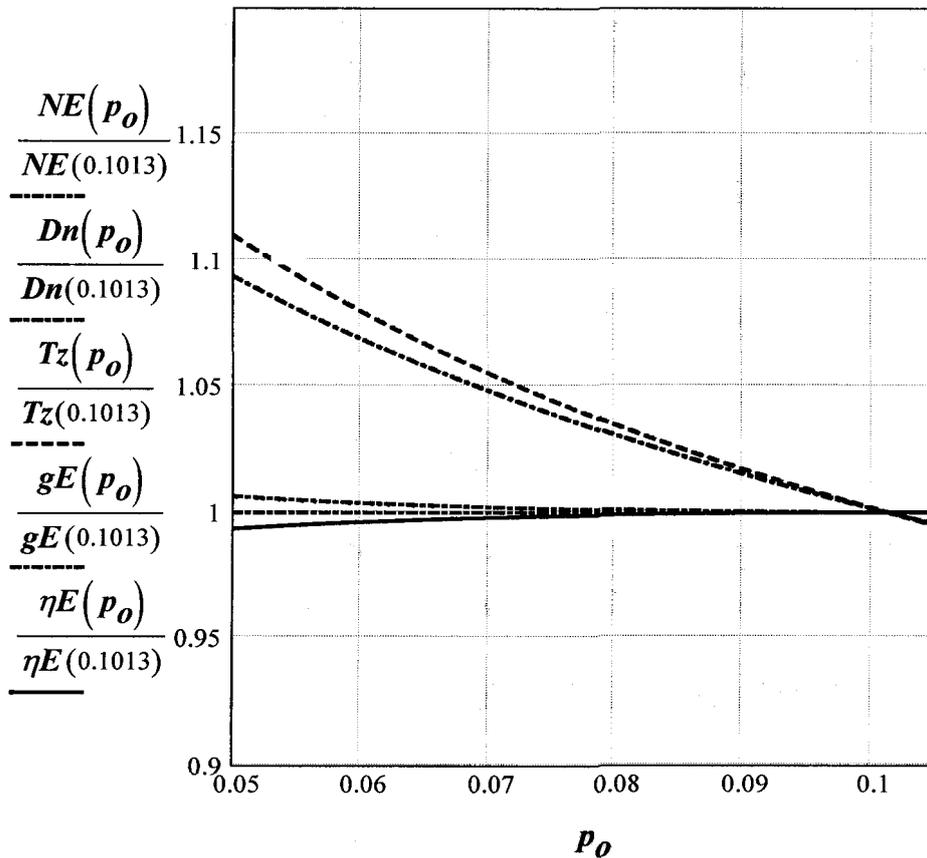


Изменение диаметра поршня в зависимости от начального давления воздуха

Рис. 7.2. Программа R_PO_1.mcd (продолжение)



Изменение мощности дизеля в зависимости от начального давления



Относительное изменение параметров цикла дизеля при изменении начального давления воздуха при $Ne = \text{const}$

Рис. 7.2. Программа R_P0_1.mcd (окончание)

ПОДПРОГРАММА теплового расчета дизеля $N=VAR$ при изменении начального давления

☐ *Reference: E:\Тепловой расчет дизеля высокогорье\Pr_HCD_2.mcd*

$$NeVARPO(p_0) := NEVAR(\alpha, p_k, p_0, T_0)$$

$$MoutPO := \left\{ \begin{array}{l} p_0 \leftarrow 0.05 \\ Vout \leftarrow NeVARPO(p_0) \\ \text{for } \varepsilon \in 1..11 \\ \quad Vout \leftarrow stack(Vout, NeVARPO(p_0 + \varepsilon \cdot 0.005)) \\ Vout \end{array} \right.$$

Формат матрицы выходных данных

$$\left(\alpha \quad p_k \quad p_0 \quad T_0 \quad \eta_e \quad g_e \quad p_e \quad T_z \quad h_u \quad \eta_i \quad p_i \quad D \quad S \quad N_e \quad V_h \right)$$

Формирование векторов выходных параметров:

$$\text{КПД эффективный } \eta E := MoutPO \langle 4 \rangle ;$$

$$\text{коэффициент избытка воздуха } ПО := MoutPO \langle 2 \rangle ;$$

$$\text{удельный эффективный расход топлива } gE := MoutPO \langle 5 \rangle ;$$

$$\text{температура } Tz \quad TZ := MoutPO \langle 7 \rangle ;$$

$$\text{диаметр поршня } D \quad DP := MoutPO \langle 11 \rangle ;$$

$$\text{эффективная мощность } Ne \quad NE := MoutPO \langle 13 \rangle$$

$$\eta E(p_0) := interp(lspline(ПО, \eta E), ПО, \eta E, p_0)$$

$$gE(p_0) := interp(lspline(ПО, gE), ПО, gE, p_0)$$

$$Tz(p_0) := interp(lspline(ПО, TZ), ПО, TZ, p_0)$$

$$Dn(p_0) := interp(lspline(ПО, DP), ПО, DP, p_0)$$

$$NE(p_0) := interp(lspline(ПО, NE), ПО, NE, p_0)$$

Рис. 7.3. Подпрограмма *Pr_PO_NVR_1.mcd*

Считывание данных матриц

$M_1 := \text{READPRN}("C:\text{Мои документы}\text{TPDz}\text{MTR_HCDz.prn}")$

$V_f := \text{READPRN}("C:\text{Мои документы}\text{TPDz}\text{MTR_fuel.prn}")$

☞ *Reference:C:\Мои документы\TPDz\Pr_fuile.mcd*

☞ *Reference:C:\Мои документы\TPDz\Pr_PO_NVR_1.mcd*

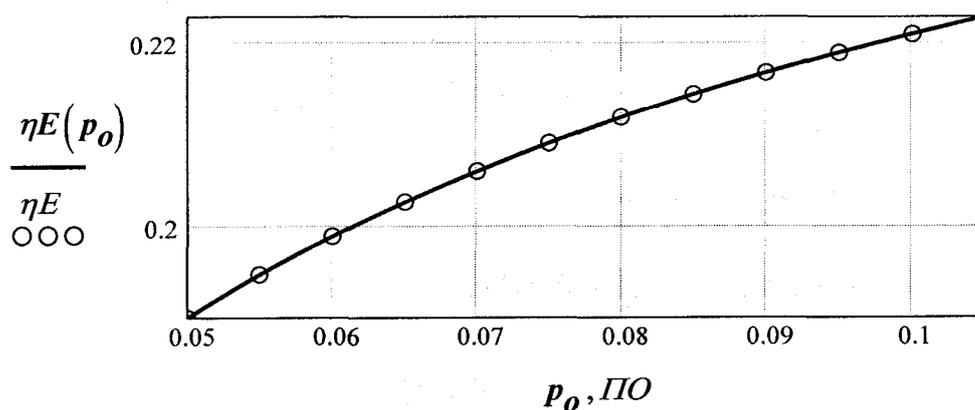
$MoutPO =$

	0	1	2	3	4	5	6
0	1.6	0.18	0.05	293	0.19	423.42	0.476
1	1.6	0.18	0.055	293	0.195	413.307	0.505
2	1.6	0.18	0.06	293	0.199	404.727	0.532
3	1.6	0.18	0.065	293	0.203	397.339	0.558
4	1.6	0.18	0.07	293	0.206	390.896	0.583
5	1.6	0.18	0.075	293	0.209	385.217	0.607
6	1.6	0.18	0.08	293	0.212	380.167	0.63
7	1.6	0.18	0.085	293	0.214	375.642	0.653
8	1.6	0.18	0.09	293	0.217	371.558	0.674
9	1.6	0.18	0.095	293	0.219	367.851	0.695
10	1.6	0.18	0.1	293	0.221	364.469	0.716
11	1.6	0.18	0.105	293	0.223	361.367	0.736

Формат матрицы

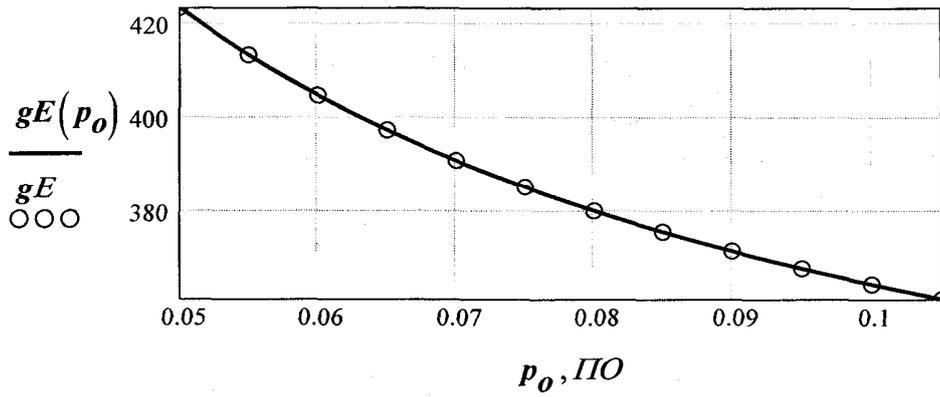
$(\alpha \ p_k \ p_0 \ T_0 \ \eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ N_e \ V_h)$;

результат теплового расчета дизеля $MTR_PO_NeVAR := MoutPO$

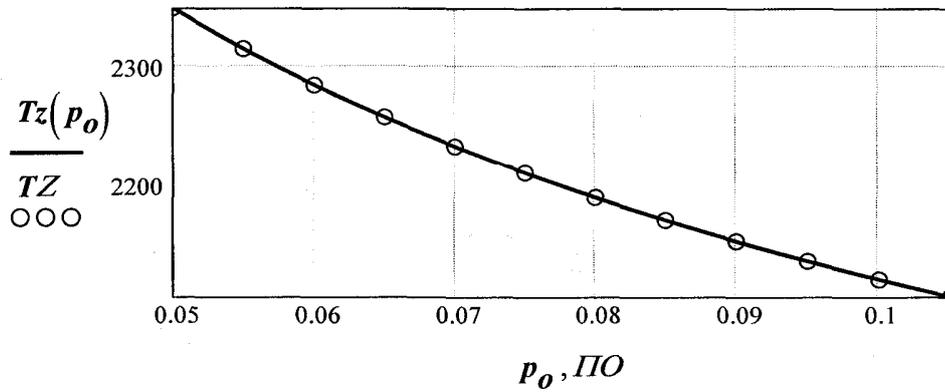


Изменение эффективного КПД в зависимости от начального давления воздуха

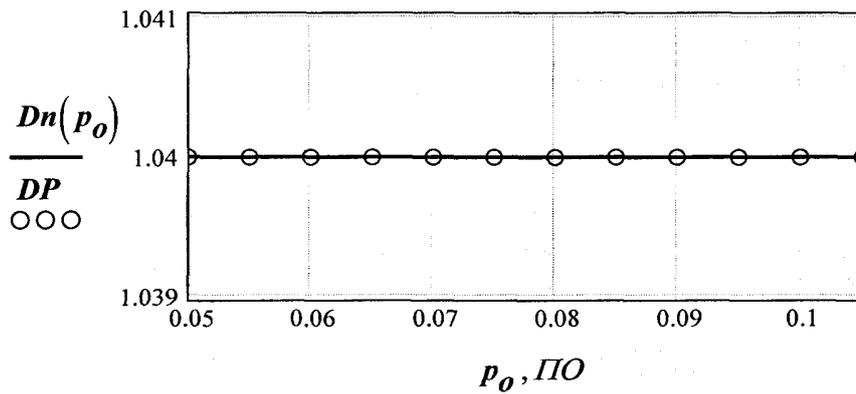
Рис. 7.4. Подпрограмма *R_PO_NVR_1.mcd* (начало)



Изменение эффективного расхода топлива в зависимости от начального давления воздуха

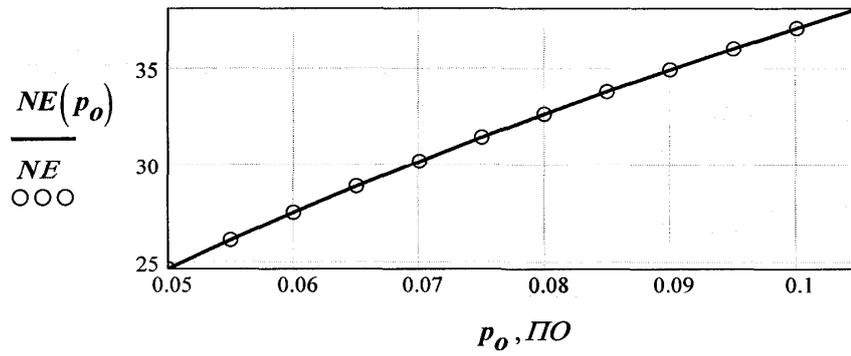


Изменение максимальной температуры в зависимости от начального давления воздуха

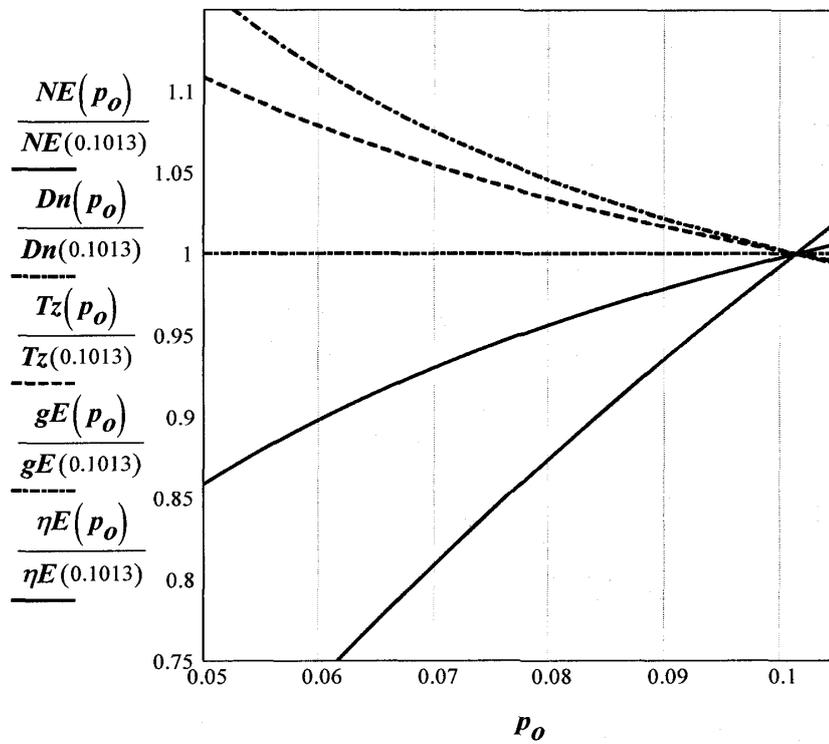


Изменение диаметра поршня дизеля в зависимости от начального давления воздуха

Рис. 7.4. Подпрограмма *R_P0_NVR_1.mcd* (продолжение)



Изменение мощности дизеля в зависимости от начального давления воздуха



Относительное изменение параметров цикла дизеля при изменении начального давления воздуха

Рис. 7.4. Подпрограмма *R_P0_NVR_1.mcd* (окончание)

ПОДПРОГРАММА

теплового расчета дизеля $N=const$ при изменении начальной температуры воздуха

☞ *Reference: E:\Тепловой расчет дизеля высокогорье\Pr_HCD_1.mcd*

$HCDT0(T_0) := HD(\alpha, p_k, p_0, T_0)$

$MoutT0 :=$ $\left\{ \begin{array}{l} T_0 \leftarrow 233 \\ Vout \leftarrow HCDT0(T_0) \\ \text{for } \varepsilon \in 1, 2..12 \\ \quad Vout \leftarrow stack(Vout, HCDT0(T_0 + \varepsilon \cdot 10)) \\ Vout \end{array} \right.$

Формат матрицы выходных данных

$(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \ N_e \ T_0)$

Формирование векторов выходных параметров:

КПД эффективный $\eta E := MoutT0^{(0)}$;

коэффициент избытка воздуха $Alfa := MoutT0^{(10)}$;

удельный эффективный расход топлива $gE := MoutT0^{(1)}$;

температура $Tz \ TZ := MoutT0^{(3)}$;

диаметр поршня $D \ DP := MoutT0^{(7)}$;

эффективная мощность $Ne \ NE := MoutT0^{(11)}$

начальная температура $TO := MoutT0^{(12)}$.

$Dn(TO) := interp(lspline(TO, DP), TO, DP, TO)$

$NE(TO) := interp(lspline(TO, NE), TO, NE, TO)$

$Tz(TO) := interp(lspline(TO, TZ), TO, TZ, TO)$

$gE(TO) := interp(lspline(TO, gE), TO, gE, TO)$

$\eta E(TO) := interp(lspline(TO, \eta E), TO, \eta E, TO)$

Рис. 7.5. Подпрограмма *Pr_T0_1.mcd*

ПРОГРАММА

теплового расчета дизеля $N=const$

Считывание данных матриц

$M_1 := READPRN("C:\Мои документы\TPDz\MTR_HCDz.prn")$

$V_f := READPRN("C:\Мои документы\TPDz\MTR_fuel.prn")$

☞ *Reference:C:\Мои документы\TPDz\Pp_fuile.mcd*

☞ *Reference:E:\TPDz\Pp_T0.mcd*

$MoutT0 =$

	0	1	2	3	4	5	6
0	0.381	211.459	1.034	$2.003 \cdot 10^3$	$4.471 \cdot 10^4$	0.438	1.189
1	0.381	211.224	0.994	$2.03 \cdot 10^3$	$4.471 \cdot 10^4$	0.438	1.143
2	0.382	211.048	0.957	$2.056 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	1.101
3	0.382	210.924	0.923	$2.083 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	1.061
4	0.382	210.848	0.891	$2.109 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	1.024
5	0.382	210.813	0.86	$2.136 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.989
6	0.382	210.817	0.832	$2.162 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.956
7	0.382	210.856	0.805	$2.189 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.926
8	0.382	210.926	0.78	$2.215 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.897
9	0.382	211.024	0.756	$2.241 \cdot 10^3$	$4.471 \cdot 10^4$	0.439	0.869

Формат матрицы $(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \ N_e \ T_0)$;

результат теплового расчета дизеля $MTR_HCD_T0 := MoutT0$

Запись результатов расчета в структурированный файл:

$WRITEPRN("C:\Мои документы\TPDz\MTR_HCD_T0.prn") := MTR_HCD_T0$

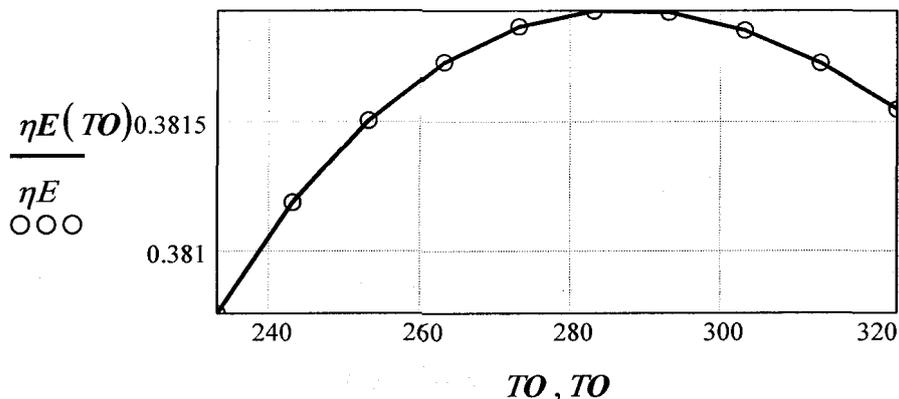
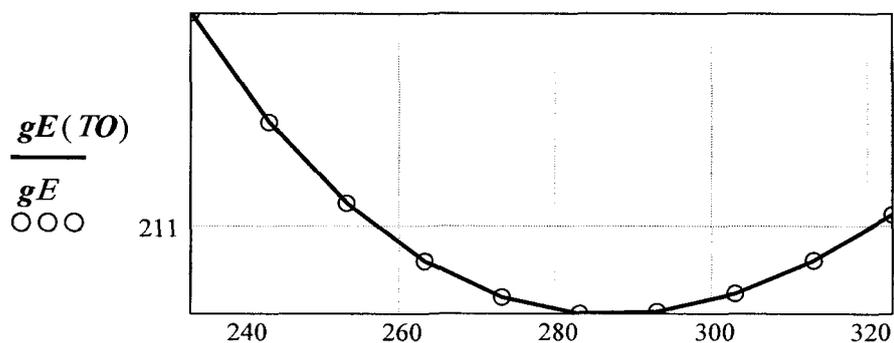
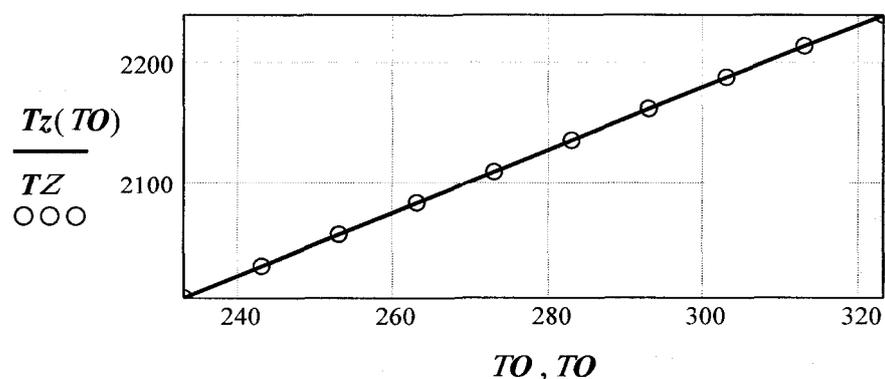


Рис. Изменение эффективного КПД в зависимости от начальной температуры воздуха

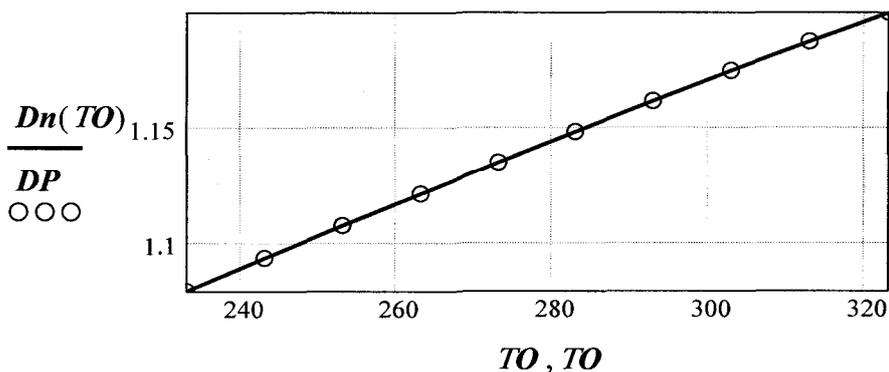
Рис. 7.6. Программа $R_T0_1.mcd$ (начало)



Изменение эффективного расхода топлива в зависимости от начальной температуры воздуха



Изменение максимальной температуры в зависимости от начальной температуры воздуха



Изменение диаметра поршня дизеля в зависимости от начальной температуры воздуха

Рис. 7.6. Программа *R_TO_1.mcd* (продолжение)



Pp_P0_1.mcd. Результатом работы программы *R_T0_1.mcd* является расчет параметров цикла дизеля (прямая задача) при изменении начальной температуры и графики зависимости параметров дизеля в функции от температуры атмосферы.

Подпрограмма *Pp_T0_NVR_1.mcd* и подпрограмма *R_T0_NVR_1.mcd* (рис. 7.7, рис. 7.8) структурно и информационно организованы аналогично программным модулям *Pp_T0_1.mcd* и *R_T0_1.mcd* с учетом того, что они решают обратную задачу.

ПОДПРОГРАММА теплового расчета дизеля $N=var$ при изменении начальной температуры воздуха

$M_1 := READPRN("MTR_HCDz_WHC.prm")$

$\alpha := M_{1_{0,0}}$ $\xi_{ex} := M_{1_{1,0}}$ $\varepsilon := M_{1_{1,4}}$ $i_u := M_{1_{2,4}}$ $\Delta T_x := M_{1_{0,3}}$
 $\eta_{ka} := M_{1_{0,4}}$ $p_k := M_{1_{0,1}}$ $p_\theta := M_{1_{0,5}}$ $p_{kr} := M_{1_{1,1}}$ $n_1 := M_{1_{1,5}}$
 $p_x := M_{1_{0,2}}$ $\Delta T := M_{1_{1,2}}$ $p_z := M_{1_{1,6}}$ $\xi_z := M_{1_{1,7}}$ $T_r := M_{1_{1,3}}$
 $k := M_{1_{0,7}}$ $\xi_{int} := M_{1_{0,8}}$ $\xi_b := M_{1_{1,8}}$ $i_{sd} := M_{1_{2,5}}$ $V_h := M_{1_{2,7}}$
 $n_{ds} := M_{1_{2,3}}$ $\eta_m := M_{1_{2,1}}$ $\phi_n := M_{1_{2,0}}$ $D := M_{1_{2,6}}$ $S := M_{1_{2,7}}$

$NeVARTO(T_0) :=$

$h_u \leftarrow H_u(V_f)$
$l_\theta \leftarrow L_\theta(V_f)$
$m_\theta \leftarrow M_\theta(V_f)$
$\alpha \leftarrow \alpha \cdot \frac{288}{T_0}$
$M_1 \leftarrow \alpha \cdot L_\theta(V_f)$
$M_2 \leftarrow M_\theta(V_f) + (\alpha - 1) \cdot L_\theta(V_f)$
$r_\theta \leftarrow \frac{M_\theta(V_f)}{[M_\theta(V_f) + (\alpha - 1) \cdot L_\theta(V_f)]}$
$r_\alpha \leftarrow \frac{[(\alpha - 1) \cdot L_\theta(V_f)]}{M_2}$
$\mu_\theta \leftarrow \frac{M_2}{M_1}$
$p'_k \leftarrow p_k + p_x$
$T_k \leftarrow T_0 \cdot \left[1 + \frac{\left(\frac{p'_k}{p_\theta} \right)^{\frac{k-1}{k}} - 1}{\eta_{ka}} \right] - \Delta T_x$
$p_a \leftarrow \xi_{int} \cdot p_k$
$p_r \leftarrow \xi_{ex} \cdot \frac{p_k}{p_{kr}}$

Рис. 7.7. Подпрограмма $Pp_T0_NVR.mcd$ (начало)

$$\gamma \leftarrow \left(\frac{T_k + \Delta T}{T_r} \right) \cdot \frac{P_r}{\varepsilon \cdot P_a - P_r}$$

$$T_a \leftarrow \frac{T_k + \Delta T + \gamma \cdot T_r}{1 + \gamma}$$

$$\eta_v \leftarrow \frac{\varepsilon}{\varepsilon - 1} \cdot \frac{P_a}{P_k} \cdot \frac{T_k}{(T_k + \Delta T)} \cdot \left(1 - \frac{P_r}{\varepsilon \cdot P_a} \right)$$

$$P_c \leftarrow P_a \cdot \varepsilon^{n_1}$$

$$T_c \leftarrow T_a \cdot \varepsilon^{n_1 - 1}$$

$$\lambda \leftarrow \frac{P_z}{P_c}$$

$$\mu \leftarrow \frac{\mu_0 + \gamma}{1 + \gamma}$$

$$q_1 \leftarrow \frac{1 + \gamma \cdot r_\alpha}{1 + \gamma}$$

$$q_2 \leftarrow \frac{\gamma \cdot r_\theta}{1 + \gamma}$$

$$U_{лев} \leftarrow \xi_z \cdot \frac{h_u}{(1 + \gamma) \cdot M_1} + 0 \dots$$

$$+ q_1 \cdot U_{возд}(T_c - 273) \dots$$

$$+ 0 + q_2 \cdot U_{ncz}(T_c - 273) + 8.314 \cdot \lambda \cdot T_c$$

$$t \leftarrow 1700$$

$$t_z \leftarrow \text{root} \left[\begin{array}{l} U_{лев} \dots \\ + 0 - \mu \cdot \left[r_\alpha \cdot U_{возд}(t) \dots \right. \\ \left. + 0 + r_\theta \cdot U_{ncz}(t) + 8.314 \cdot (t + 273) \right] \end{array} \right], t$$

Рис. 7. 7. Подпрограмма *Pp_T0_NVR.mcd* (продолжение)

$$\begin{aligned}
T_z &\leftarrow t_z + 273 \\
\rho &\leftarrow \mu \cdot \frac{T_z}{\lambda \cdot T_c} \\
\delta &\leftarrow \frac{\varepsilon}{\rho} \\
U'' &\leftarrow (\xi_b - \xi_z) \cdot \frac{h_u}{(\mu_\theta + \gamma) \cdot M_1} \\
n &\leftarrow 1.15 \\
n_2 &\leftarrow \text{root} \left[\begin{array}{l} U'' \dots \\ + 0 - \left[8.314 \cdot \frac{(T_z - T_z \cdot \delta^{1-n})}{n-1} \right] \dots \\ + 0 - r_\alpha \cdot \left(U_{603\delta}(t_z) \dots \right. \\ \quad \left. + 0 - U_{603\delta} \left(\begin{array}{l} T_z \cdot \delta^{1-n} \dots \\ + 0 - 273 \end{array} \right) \right) \dots \\ + 0 - r_\theta \cdot \left(U_{ncz}(t_z) - U_{ncz}(T_z \cdot \delta^{1-n} - 273) \right) \end{array} \right]
\end{aligned}$$

Рис. 7. 7. Подпрограмма *Pp_T0_NVR.mcd* (продолжение)

$$P_b \leftarrow P_z \cdot \delta^{-n_2}$$

$$T_b \leftarrow T_z \cdot \delta^{1-n_2}$$

$$P_{ip} \leftarrow \frac{P_c}{\varepsilon - 1} \cdot \left[\begin{array}{l} \lambda \cdot (\rho - 1) \dots \\ + 0 + \lambda \cdot \frac{\rho}{n_2 - 1} \cdot \left(1 - \frac{T_b}{T_z} \right) \dots \\ + 0 - \frac{1}{n_1 - 1} \cdot \left(1 - \frac{T_a}{T_c} \right) \end{array} \right]$$

$$P_i \leftarrow \phi_n \cdot P_{ip}$$

$$\eta_i \leftarrow 8.314 \cdot \frac{P_i \cdot M_I \cdot T_k}{h_u \cdot \eta_v \cdot P_k}$$

$$g_i \leftarrow \frac{3600000}{h_u \cdot \eta_i}$$

$$\eta_e \leftarrow \eta_i \cdot \eta_m$$

$$P_e \leftarrow P_i \cdot \eta_m$$

$$g_e \leftarrow \frac{3600000}{h_u \cdot \eta_e}$$

$$V_h \leftarrow \pi \cdot \frac{D^2}{4} (D \cdot i_{sd})$$

$$N_e \leftarrow V_h \cdot \frac{(P_e \cdot n_{ds} \cdot i_u)}{120}$$

$$S \leftarrow D \cdot i_{sd}$$

$$\left(\alpha \quad P_k \quad P_0 \quad T_0 \quad \eta_e \quad g_e \quad P_e \quad T_z \quad h_u \quad \eta_i \quad P_i \quad D \quad S \quad N_e \quad V_h \right)$$

Рис. 7. 7. Подпрограмма *Pp_T0_NVR.mcd* (продолжение)

Формирование матрицы результатов расчета (цикл *for*)

```

MoutT0 :=  $\left\{ \begin{array}{l} T_0 \leftarrow 233 \\ V_{out} \leftarrow NeVARTO(T_0) \\ \text{for } \varepsilon \in 1, 2..12 \\ \quad V_{out} \leftarrow stack(V_{out}, NeVARTO(T_0 + \varepsilon \cdot 10)) \\ V_{out} \end{array} \right.$ 

```

Формат матрицы выходных данных

$(\alpha \quad P_k \quad P_0 \quad T_0 \quad \eta_e \quad g_e \quad P_e \quad T_z \quad h_u \quad \eta_i \quad P_i \quad D \quad S \quad N_e \quad V_h)$

Формирование векторов выходных данных:

$\eta E := MoutT0^{(4)} \quad T O := MoutT0^{(3)} \quad g E := MoutT0^{(5)} \quad T Z := MoutT0^{(7)}$
 $DP := MoutT0^{(11)} \quad NE := MoutT0^{(13)}$

Интерполяция результатов расчета кубическими сплайнами:

$\eta E(T_H) := interp(lspline(T O, \eta E), T O, \eta E, T_H)$

$g E(T_H) := interp(lspline(T O, g E), T O, g E, T_H)$

$T z(T_H) := interp(lspline(T O, T Z), T O, T Z, T_H)$

$Dn(T_H) := interp(lspline(T O, DP), T O, DP, T_H)$

$NE(T_H) := interp(lspline(T O, NE), T O, NE, T_H)$

Конец

подпрограммы теплового расчета дизеля *Pp_T0_NVR.mcd*

Рис. 7. 7. Подпрограмма *Pp_T0_NVR.mcd* (окончание)

ПРОГРАММА

теплового расчета дизеля $N=VAR$ при изменении начальной температуры воздуха

Считывание данных матриц:

$M_f := READPRN("MTR_HCDz_WHC.pm")$

$V_f := READPRN("MTR_fuel.pm")$

→ *Reference:C:\Мои документы\TPDz\HCDz T0 NVR\Pp_fuile.mcd*

→ *Reference:C:\Мои документы\TPDz\HCDz T0 NVR\Pp T0 NVR.mcd*

Матрица результатов расчета

	0	1	2	3	4	5	6
$MoutT0 =$	1.978	0.101	0.103	233	0.392	205.148	0.862
	1.896	0.101	0.103	243	0.391	206.174	0.86
	1.821	0.101	0.103	253	0.389	207.201	0.857
	1.752	0.101	0.103	263	0.387	208.23	0.854
	1.688	0.101	0.103	273	0.385	209.261	0.851
	1.628	0.101	0.103	283	0.383	210.294	0.848
	1.573	0.101	0.103	293	0.381	211.328	0.844
	1.521	0.101	0.103	303	0.379	212.365	0.841
	1.472	0.101	0.103	313	0.377	213.403	0.838
	1.427	0.101	0.103	323	0.375	214.444	0.835
	1.384	0.101	0.103	333	0.374	215.487	0.831
	1.343	0.101	0.103	343	0.372	216.531	0.828
	1.305	0.101	0.103	353	0.37	217.578	0.825

Формат матрицы

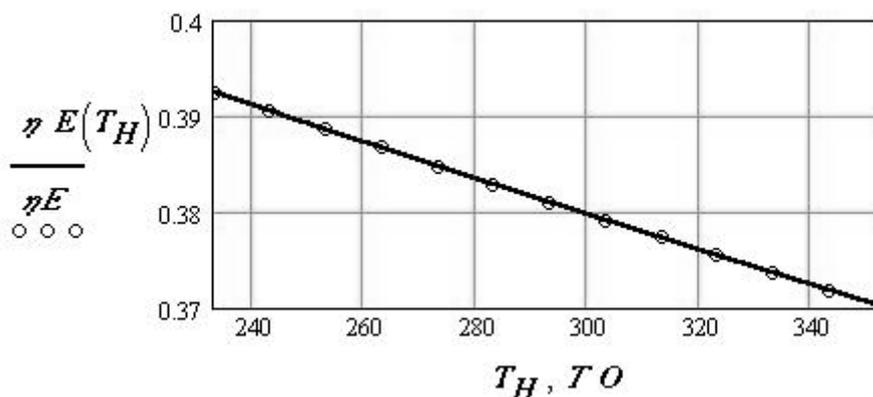
$(\alpha \quad p_k \quad p_0 \quad T_0 \quad \eta_e \quad g_e \quad p_e \quad T_z \quad h_u \quad \eta_i \quad p_i \quad D \quad S \quad N_e \quad V_h)$

Результат теплового расчета дизеля

$MTR_HCD_T0 := MoutT0$

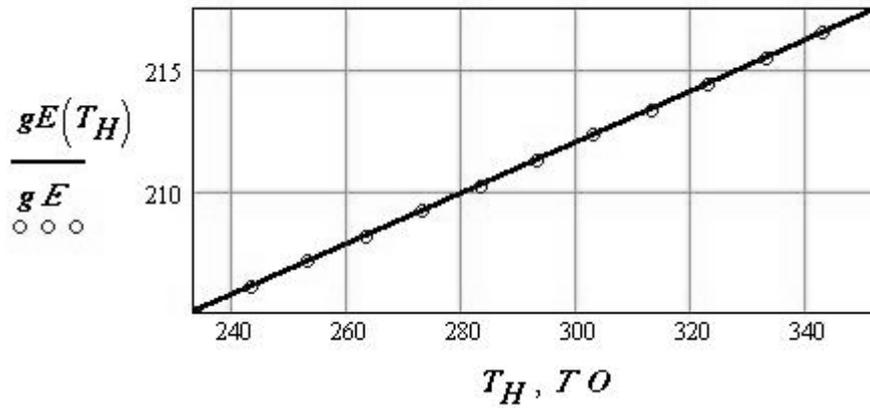
Запись матрицы выходных данных в текстовый структурированный файл:

$WRITEPRN("MTR_HCD_T0_NVR.pm") := MTR_HCD_T0$

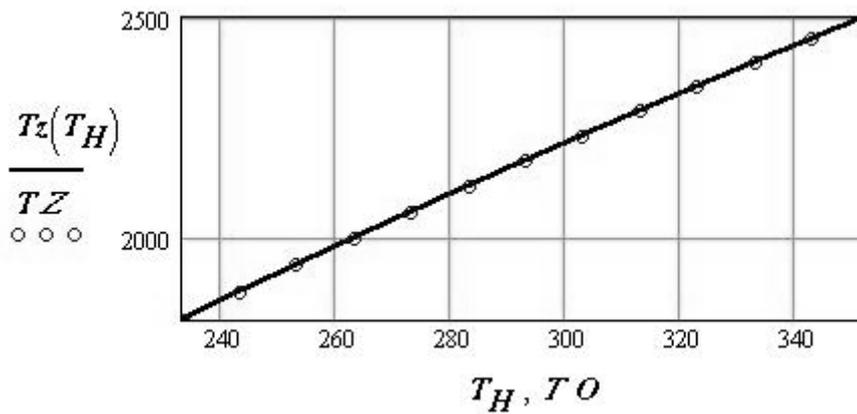


Изменение эффективного КПД в зависимости от начальной температуры Воздуха

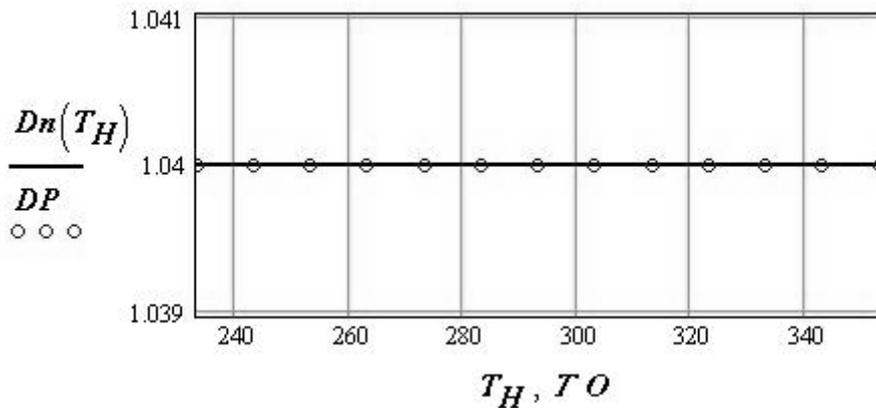
Рис. 7.8. Программа $R_T0_NVR.mcd$ (начало)



Изменение эффективного расхода топлива в зависимости от начальной температуры воздуха

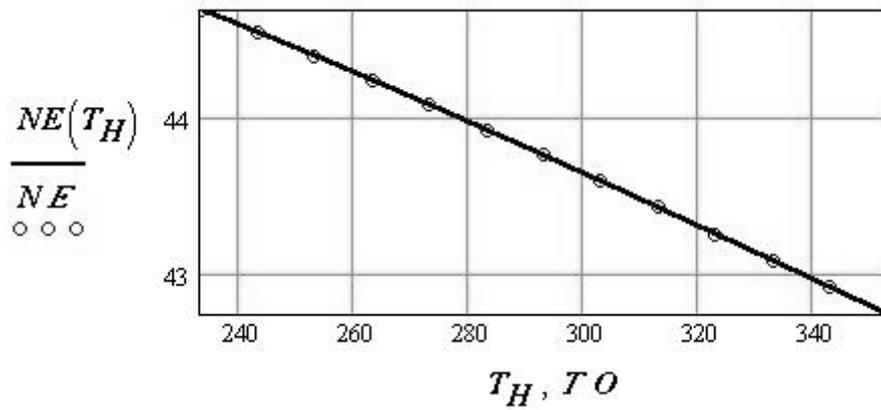


Изменение максимальной температуры в зависимости от начальной температуры воздуха



Изменение диаметра поршня дизеля в зависимости от начальной температуры воздуха

Рис. 7.8. Программа *R_T0_NVR.mcd* (продолжение)



Изменение мощности дизеля в зависимости от начальной температуры воздуха

Конец
 программы теплового расчета дизеля *R_T0_NVR.mcd*

Рис. 7.8. Программа *R_T0_NVR.mcd* (окончание)

ТЕМА 8. ПРОГРАММНЫЕ МОДУЛИ ДАВЛЕНИЯ НАДДУВА ДИЗЕЛЯ И КОЭФФИЦИЕНТА ИЗБЫТКА ВОЗДУХА

В данном разделе приведены тексты программных модулей, предназначенных для расчета параметров цикла дизеля при изменении давления наддува и коэффициента избытка воздуха при решении прямой и обратной задач проектирования.

8. 1. Блок давления наддува

Блок давления наддува включает подпрограммы *Pp_Pk.mcd*, *Pp_Pk_NVR.mcd* (рис.8.1, рис. 8.3) и *R_Pk.mcd*, *R_Pk_NVR.mcd* (рис. 8.2, рис. 8.4), которые решают прямую и обратную задачи. Общее построение этих программных модулей аналогично рассмотренным в предыдущих разделах.

8. 2. Блок коэффициента избытка воздуха

Блок коэффициента избытка воздуха включает подпрограммы *Pp_ALFA.mcd*, *Pp_ALFA_NVR.mcd* (рис. 8.5, рис. 8.7) и программы *R_ALFA.mcd*, *R_ALFA_NVR.mcd* (рис. 8,6, рис. 8.7), которые решают прямую и обратную задачи. Общее построение этих программных модулей аналогично рассмотренным в предыдущих разделах.

ПОДПРОГРАММА

теплового расчета дизеля $N=\text{const}$ при изменении давления наддува воздуха

Reference: \ Pp_HCD_1.mcd

$HDPK(p_k) := HD(\alpha, p_k, p_0, T_0)$

$MoutPK :=$ $\left\{ \begin{array}{l} p_k \leftarrow 0.11 \\ Vout \leftarrow HCDPK(p_k) \\ \text{for } \varepsilon \in 1, 2..11 \\ \quad Vout \leftarrow \text{stack}(Vout, HCDPK(p_k + \varepsilon \cdot 0.01)) \\ Vout \end{array} \right.$

Формат матрицы выходных данных

$(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \ N_e \ p_0 \ p_k)$

Формирование векторов выходных параметров:

КПД эффективный $\eta_E := MoutPK^{(0)}$;

начальное давление воздуха $PK := MoutPK^{(13)}$;

удельный эффективный расход топлива $g_E := MoutPK^{(1)}$;

температура $T_z \ TZ := MoutPK^{(3)}$;

диаметр поршня $D \ DP := MoutPK^{(7)}$;

эффективная мощность $N_e \ NE := MoutPK^{(11)}$.

$\eta_E(p_k) := \text{interp}(lspline(PK, \eta_E), PK, \eta_E, p_k)$

$g_E(p_k) := \text{interp}(lspline(PK, g_E), PK, g_E, p_k)$

$T_z(p_k) := \text{interp}(lspline(PK, TZ), PK, TZ, p_k)$

$Dn(p_k) := \text{interp}(lspline(PK, DP), PK, DP, p_k)$

$NE(p_k) := \text{interp}(lspline(PK, NE), PK, NE, p_k)$

Конец подпрограммы теплового расчета дизеля

Pp_PK.mcd

Рис. 8. 1. Подпрограмма Pp_Pk.mcd

ПРОГРАММА

теплового расчета дизеля $N=\text{const}$

Считывание данных матриц

$M_1 := \text{READPRN}(\text{"C:\MTR_HCDz.prn"})$

$V_f := \text{READPRN}(\text{"C:\MTR_fuel.prn"})$

☒ *Reference:C:\Pp_fuile.mcd*

☒ *Reference:C:\Pp_PK.mcd*

$MoutPK =$

	7	8	9	10	11	12	13
0	1.144	1.258	1.293	1.6	60	0.103	0.11
1	1.127	1.239	1.235	1.6	60	0.103	0.12
2	1.111	1.222	1.184	1.6	60	0.103	0.13
3	1.097	1.207	1.14	1.6	60	0.103	0.14
4	1.085	1.193	1.102	1.6	60	0.103	0.15
5	1.073	1.181	1.067	1.6	60	0.103	0.16
6	1.063	1.169	1.037	1.6	60	0.103	0.17
7	1.053	1.159	1.01	1.6	60	0.103	0.18
8	1.045	1.149	0.985	1.6	60	0.103	0.19
9	1.037	1.141	0.962	1.6	60	0.103	0.2
10	1.029	1.132	0.942	1.6	60	0.103	0.21
11	1.023	1.125	0.924	1.6	60	0.103	0.22

Фрагмент матрицы

Формат матрицы

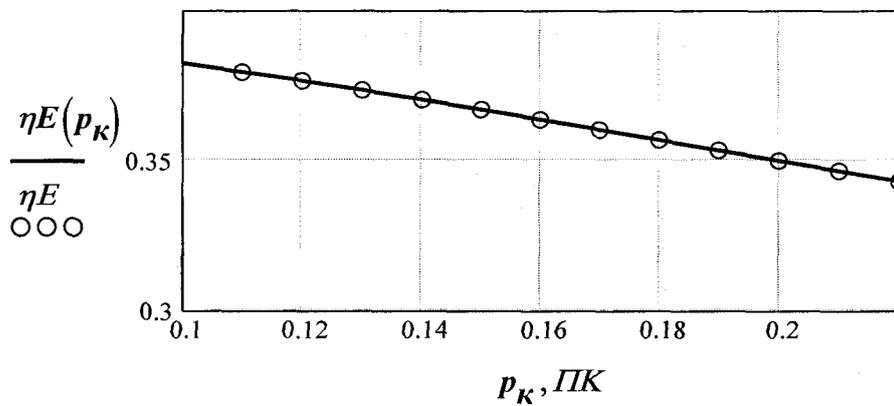
$(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \ N_e \ p_0 \ p_k)$;

результат теплового расчета дизеля $MTR_HCD_PK := MoutPK$

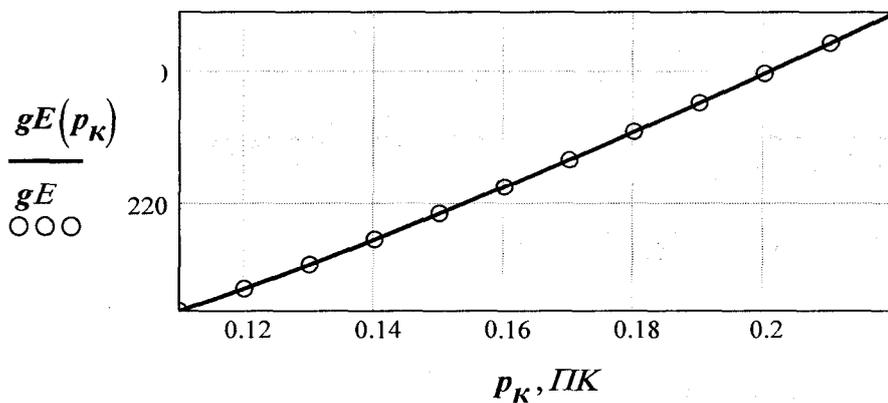
Запись результатов расчета в структурированный файл:

$WRITEPRN(\text{"C:\Мои документы\TPDz\MTR_HCD_PK.prn"}) := MTR_HCD_PK$

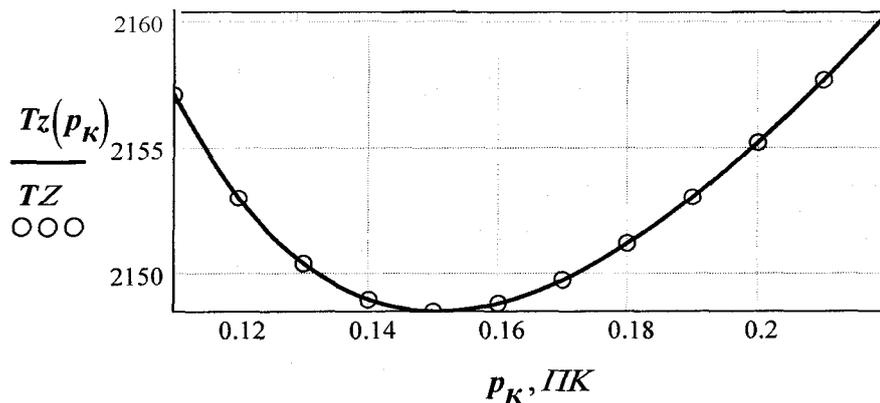
Рис. 8. 2. Программа *R_Pk.mcd* (начало)



Изменение эффективного КПД в зависимости от давления наддува

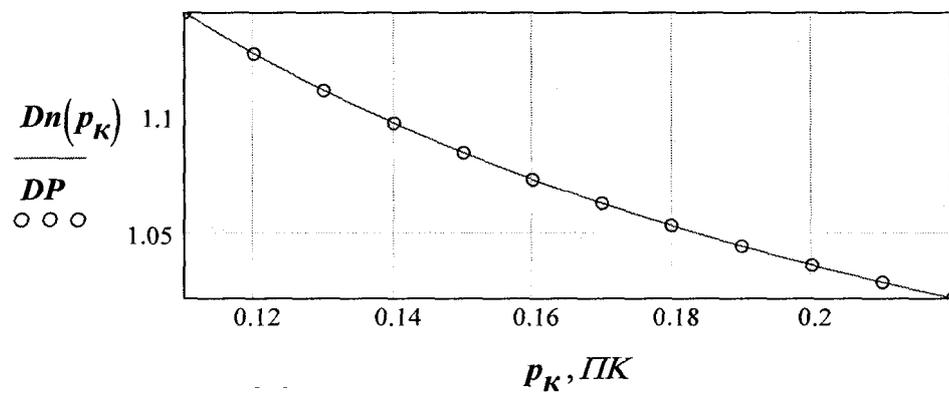


Изменение эффективного расхода топлива в зависимости от давления наддува

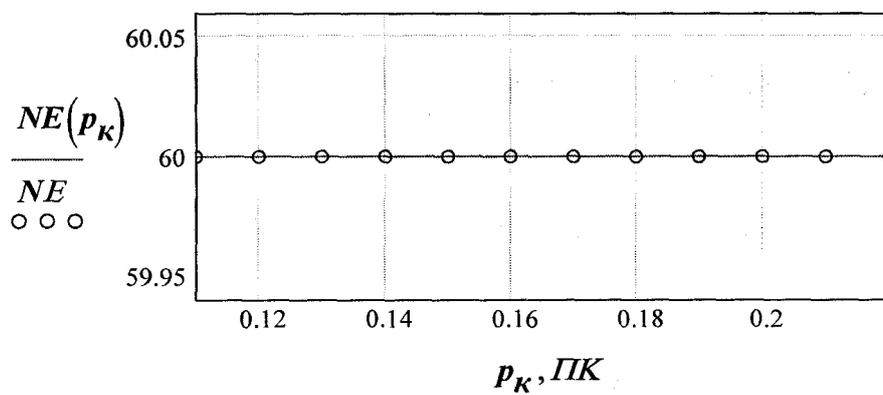


Изменение максимальной температуры в зависимости от давления наддува дизеля

Рис. 8.2. Программа R_Pk.mcd (продолжение)

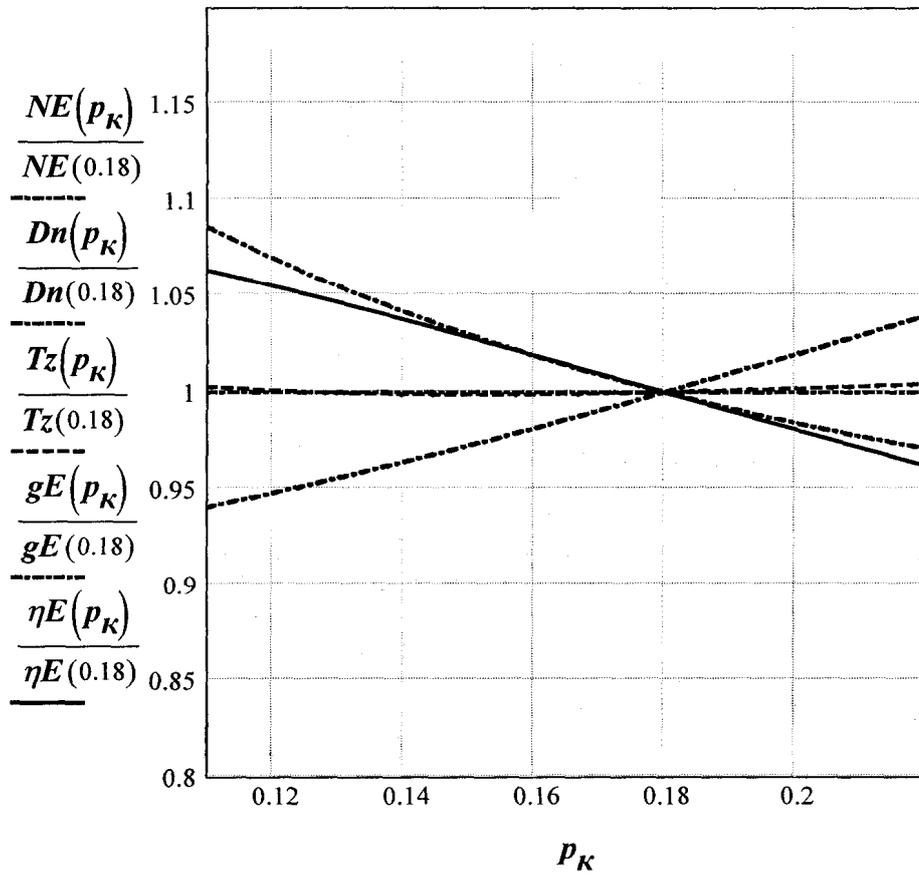


Изменение диаметра поршня двигателя в зависимости от давления наддува



Изменение мощности дизеля в зависимости от давления наддува

Рис. 8.2. Программа *R_Pk.mcd* (продолжение)



Относительное изменение параметров цикла дизеля при изменении давления наддува при $Ne=\text{const}$

Рис. 8.2. Программа *R_Pk.mcd* (окончание)

ПОДПРОГРАММА

теплового расчета дизеля $N=VAR$ при изменении давления наддува

☒ *Reference: E:\Тепловой расчет дизеля высокогорье\Pr_HCD_2.mcd*

$NeVARPK(p_k) := NEVAR(\alpha, p_k, p_0, T_0)$

$MoutPK :=$

$p_k \leftarrow 0.11$
$Vout \leftarrow NeVARPK(p_k)$
$for \ \varepsilon \in 1, 2 .. 11$
$Vout \leftarrow stack(Vout, NeVARPK(p_k + \varepsilon \cdot 0.01))$
$Vout$

Формат матрицы выходных данных

$(\alpha \ p_k \ p_0 \ T_0 \ \eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ N_e \ V_h)$

Формирование векторов выходных параметров:

КПД эффективный $\eta E := MoutPK \langle 4 \rangle$;

давление наддува $PK := MoutPK \langle 1 \rangle$;

удельный эффективный расход топлива $gE := MoutPK \langle 5 \rangle$;

температура $Tz \ TZ := MoutPK \langle 7 \rangle$;

диаметр поршня $D \ DP := MoutPK \langle 11 \rangle$;

эффективная мощность $Ne \ NE := MoutPK \langle 13 \rangle$.

$\eta E(p_k) := interp(lspline(PK, \eta E), PK, \eta E, p_k)$

$gE(p_k) := interp(lspline(PK, gE), PK, gE, p_k)$

$Tz(p_k) := interp(lspline(PK, TZ), PK, TZ, p_k)$

$Dn(p_k) := interp(lspline(PK, DP), PK, DP, p_k)$

$NE(p_k) := interp(lspline(PK, NE), PK, NE, p_k)$

Конец подпрограммы теплового расчета дизеля

Pr_PK_NVR.mcd

Рис. 8.3. Подпрограмма *Pr_Pk_NVR.mcd*

ПРОГРАММА

теплового расчета дизеля $N=VAR$ при изменении давления наддува

Считывание данных матриц

$M_1 := READPRN("C:\Мои документы\TPDz\MTR_HCDz.prn")$

$V_f := READPRN("C:\Мои документы\TPDz\MTR_fuel.prn")$

☞ *Reference:C:\Мои документы\TPDz\Pr_fuile.mcd*

☞ *Reference:C:\Мои документы\TPDz\Pr_PK_NVR.mcd*

$MoutPK =$

	0	1	2	3	4	5	6
0	1.6	0.11	0.103	293	0.38	212.131	0.87
1	1.6	0.12	0.103	293	0.377	213.775	0.911
2	1.6	0.13	0.103	293	0.374	215.536	0.95
3	1.6	0.14	0.103	293	0.37	217.393	0.987
4	1.6	0.15	0.103	293	0.367	219.33	1.021
5	1.6	0.16	0.103	293	0.364	221.337	1.054
6	1.6	0.17	0.103	293	0.36	223.406	1.085
7	1.6	0.18	0.103	293	0.357	225.53	1.114
8	1.6	0.19	0.103	293	0.354	227.704	1.142
9	1.6	0.2	0.103	293	0.35	229.926	1.169
10	1.6	0.21	0.103	293	0.347	232.192	1.194
11	1.6	0.22	0.103	293	0.343	234.5	1.218

Формат матрицы

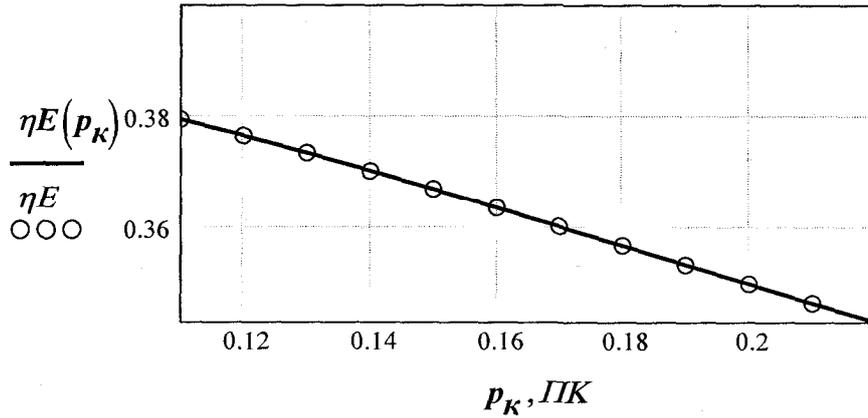
$(\alpha \ p_k \ p_0 \ T_0 \ \eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ N_e \ V_h)$;

результат теплового расчета дизеля $MTR_PK_NeVAR := MoutPK$

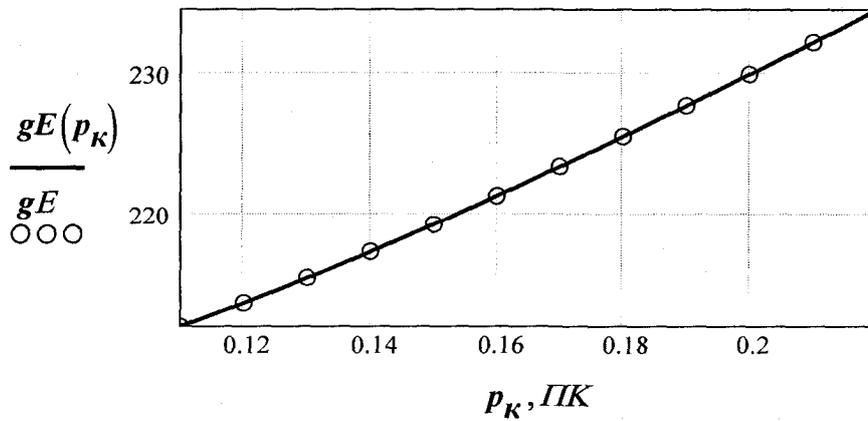
Запись результатов расчета в структурированный файл:

$WRITEPRN("C:\Мои документы\TPDz\MTR_PK_NeVAR.prn") := MTR_PK_NeVAR$

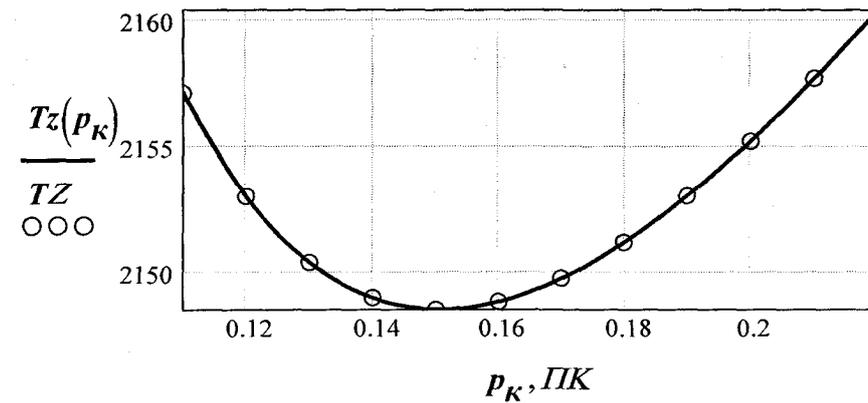
Рис. 8.4. Программа *R_Pk_NVR.mcd* (начало)



Изменение эффективного КПД в зависимости от давления наддува дизеля

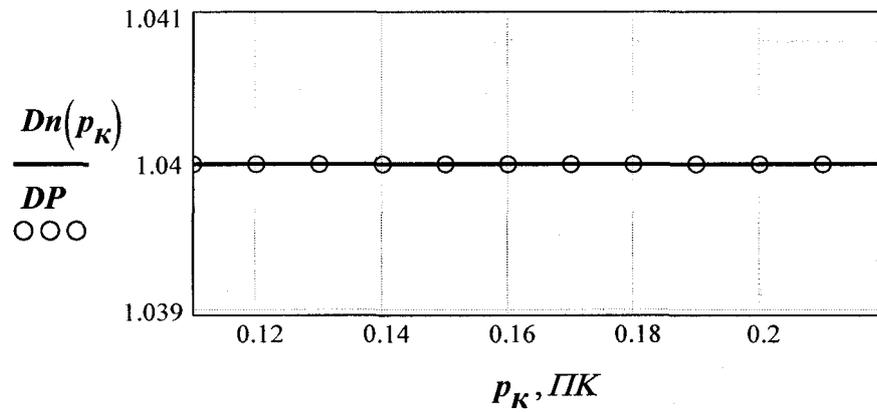


Изменение эффективного расхода топлива в зависимости от давления наддува дизеля

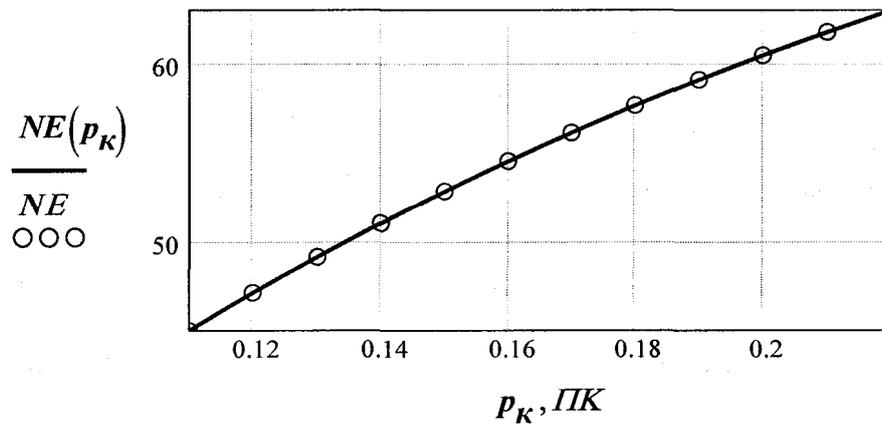


Изменение максимальной температуры в зависимости от давления наддува дизеля

Рис. 8.4. Программа *R_Pk_NVR.mcd* (продолжение)

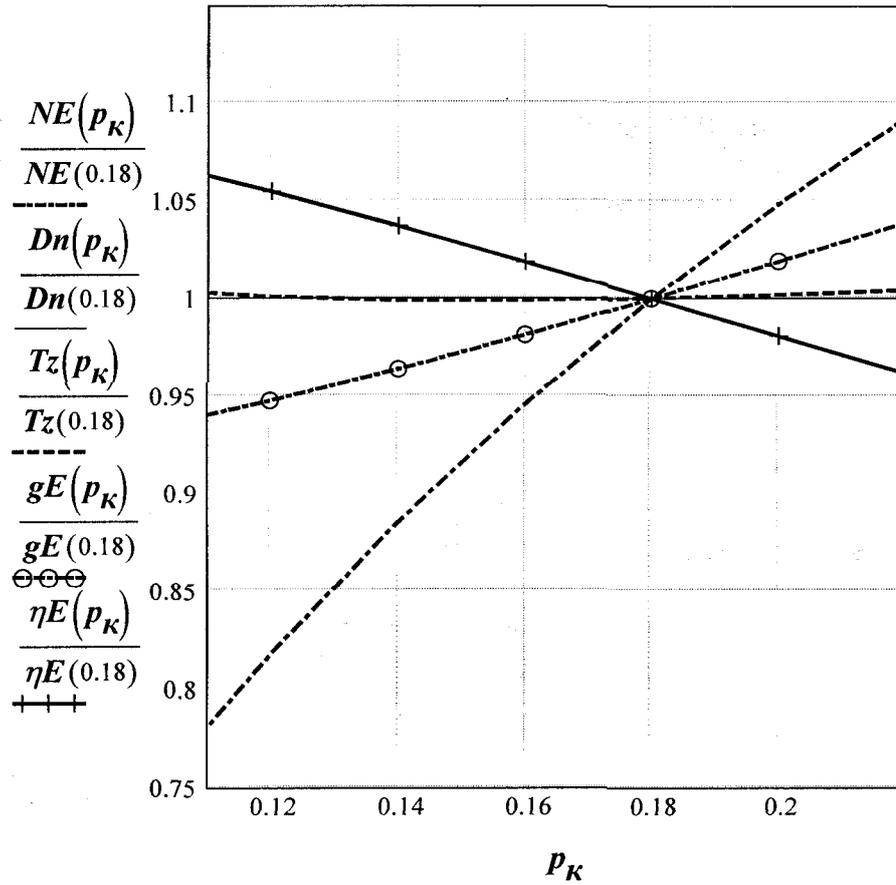


Изменение диаметра поршня в зависимости от давления наддува дизеля



Изменение мощности дизеля в зависимости от давления наддува дизеля

Рис. 8.4. Программа R_Pk_NVR.mcd (продолжение)



Относительное изменение параметров цикла дизеля при изменении давления наддува

Конец программы теплового расчета дизеля *R_PK_NVR.mcd*

Рис. 8. 10. Программа *R_Pk_NVR.mcd* (окончание)

ПОДПРОГРАММА

теплового расчета дизеля $N=\text{const}$ при изменении коэффициента избытка воздуха

```

Z := | m ← 0
      | for s ∈ 1, 1.2 .. 2.6
      |   | Dm ← HCDALFA(s)
      |   | Am ← (Dm)0,10
      |   | Bm ← (Dm)0,0
      |   | Cm ← (Dm)0,3
      |   | Em ← (Dm)0,7
      |   | Fm ← (Dm)0,11
      |   | Gm ← (Dm)0,1
      |   | m ← m + 1
      | W⟨0⟩ ← A
      | W⟨1⟩ ← B
      | W⟨2⟩ ← C
      | W⟨3⟩ ← E
      | W⟨4⟩ ← F
      | W⟨5⟩ ← G
      | W
  
```

$Mout\alpha := Z$

Формат матрицы $\alpha, \eta\epsilon, Tz, D, Ne, ge$

Формирование векторов выходных параметров:

КПД эффективный $\eta E := Mout\alpha^{\langle 1 \rangle}$;

коэффициент избытка воздуха $Alfa := Mout\alpha^{\langle 0 \rangle}$;

удельный эффективный расход топлива $gE := Mout\alpha^{\langle 5 \rangle}$

Рис. 8.5. Подпрограмма *Pp_ALFA.mcd* (начало)

температура Tz $TZ := Mout\alpha^{(2)}$;
 диаметр поршня D $DP := Mout\alpha^{(3)}$;
 эффективная мощность Ne $NE := Mout\alpha^{(4)}$.
 $\eta E(a) := interp(lspline(Alfa, \eta E), Alfa, \eta E, a)$
 $gE(a) := interp(lspline(Alfa, gE), Alfa, gE, a)$
 $Tz(a) := interp(lspline(Alfa, TZ), Alfa, TZ, a)$
 $Dn(a) := interp(lspline(Alfa, DP), Alfa, DP, a)$
 $NE(a) := interp(lspline(Alfa, NE), Alfa, NE, a)$
Конец подпрограммы теплового расчета дизеля
Pp_ALFA.mcd

Рис. 8.5. Подпрограмма *Pp_ALFA.mcd* (окончание)

ПРОГРАММА

теплового расчета дизеля при $N=\text{const}$ изменении
коэффициента избытка воздуха

Считывание данных матриц

$M_1 := \text{READPRN}(\text{"C:\Мои документы\TPDz\MTR_HCDz.prn"})$

$V_f := \text{READPRN}(\text{"C:\Мои документы\TPDz\MTR_fuel.prn"})$

☞ *Reference:C:\Мои документы\TPDz\Pr_fuile.mcd*

☞ *Reference:C:\Мои документы\TPDz\Pr_ALFA.mcd*

$$Mout\alpha = \begin{pmatrix} 1 & 0.351 & 2.655 \times 10^3 & 1.022 & 60 & 229.61 \\ 1.2 & 0.364 & 2.447 \times 10^3 & 1.072 & 60 & 221.037 \\ 1.4 & 0.374 & 2.288 \times 10^3 & 1.118 & 60 & 215.103 \\ 1.6 & 0.382 & 2.162 \times 10^3 & 1.161 & 60 & 210.817 \\ 1.8 & 0.388 & 2.06 \times 10^3 & 1.202 & 60 & 207.629 \\ 2 & 0.392 & 1.976 \times 10^3 & 1.24 & 60 & 205.209 \\ 2.2 & 0.396 & 1.905 \times 10^3 & 1.276 & 60 & 203.348 \\ 2.4 & 0.399 & 1.845 \times 10^3 & 1.31 & 60 & 201.906 \\ 2.6 & 0.401 & 1.793 \times 10^3 & 1.343 & 60 & 200.788 \end{pmatrix}$$

Формат матрицы

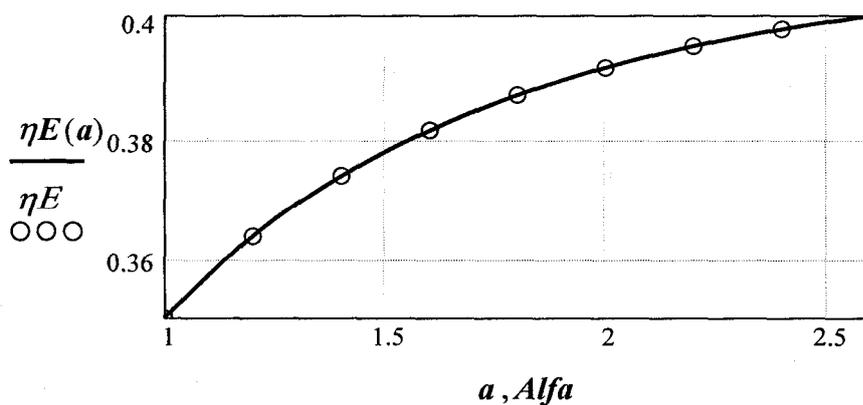
$(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha) ;$

результат теплового расчета дизеля $MTR_ALFA := Mout\alpha$

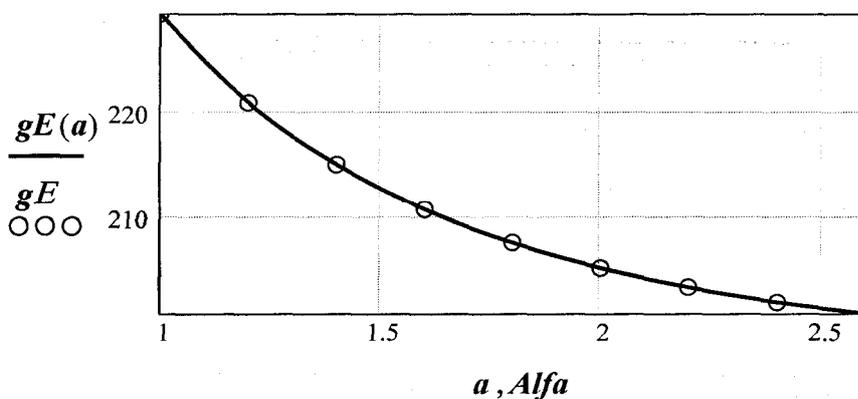
Запись результатов расчета в структурированный файл:

$WRITEPRN(\text{"C:\MTR_ALFA.prn"}) := MTR_ALFA$

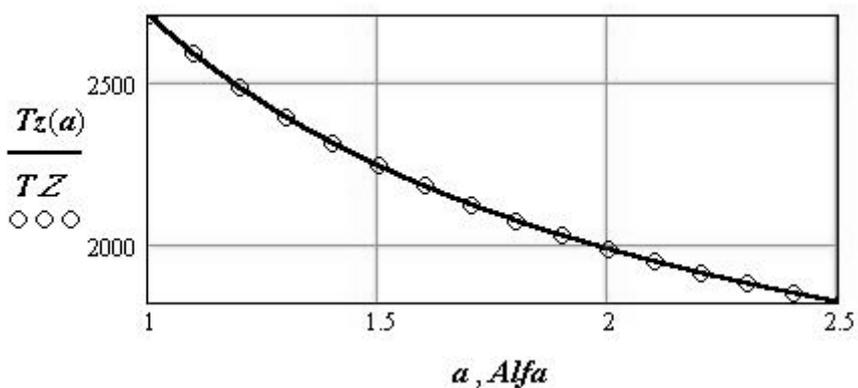
Рис. 8.6. Программа $R_ALFA.mcd$ (начало)



Изменение эффективного КПД в зависимости от коэффициента избытка воздуха

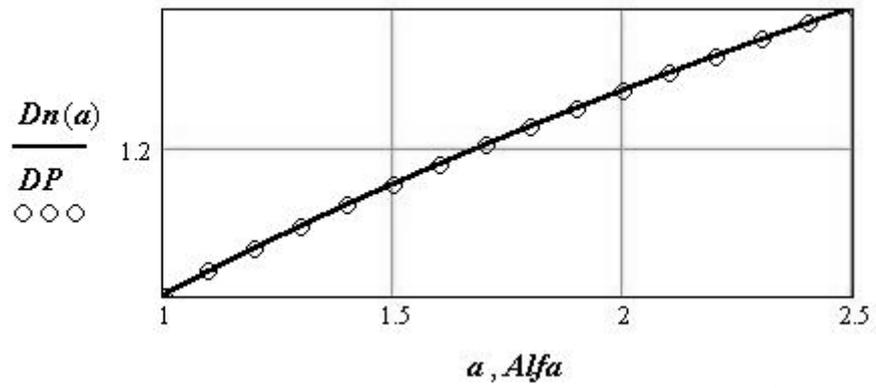


Изменение эффективного расхода топлива в зависимости от коэффициента избытка воздуха

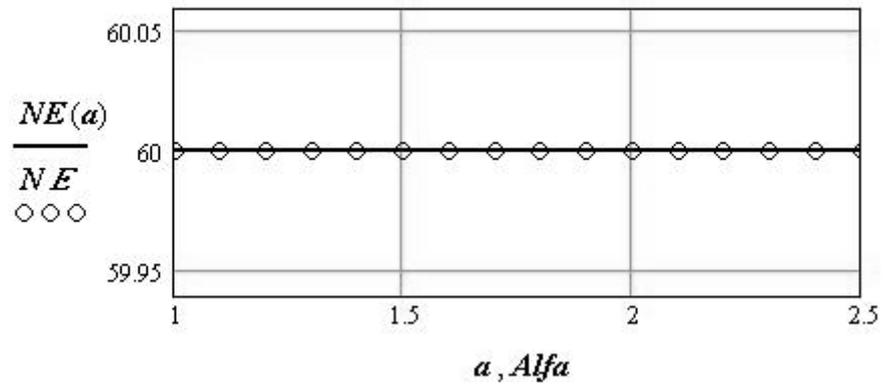


Изменение максимальной температуры в зависимости от коэффициента избытка воздуха

Рис. 8.6. Программа *R_ALFA.mcd* (продолжение)



Изменение диаметра поршня дизеля в зависимости от коэффициента избытка воздуха



Изменение мощности дизеля в зависимости от коэффициента избытка воздуха

Рис. 8.6. Программа *R_ALFA.mcd* (окончание)

ПОДПРОГРАММА

теплового расчета дизеля $N=VAR$ при изменении коэффициента избытка воздуха

$$\begin{aligned}
 NeVARALFA(\alpha) := & \left[\begin{aligned}
 h_u &\leftarrow H_u(V_f) \\
 l_0 &\leftarrow L_0(V_f) \\
 m_0 &\leftarrow M_0(V_f) \\
 M_1 &\leftarrow \alpha \cdot L_0(V_f) \\
 M_2 &\leftarrow M_0(V_f) + (\alpha - 1) \cdot L_0(V_f) \\
 r_0 &\leftarrow \frac{M_0(V_f)}{[M_0(V_f) + (\alpha - 1) \cdot L_0(V_f)]} \\
 r_\alpha &\leftarrow \frac{[(\alpha - 1) \cdot L_0(V_f)]}{M_2} \\
 \mu_0 &\leftarrow \frac{M_2}{M_1} \\
 p'_k &\leftarrow p_k + p_x \\
 T_k &\leftarrow T_0 \cdot \left[1 + \frac{\left(\frac{p'_k}{p_0} \right)^{\frac{k-1}{k}} - 1}{\eta_{ka}} \right] - \Delta T_x \\
 p_a &\leftarrow \xi_{int} \cdot p_k \\
 p_r &\leftarrow \xi_{ex} \cdot \frac{p_k}{p_{kr}} \\
 \gamma &\leftarrow \left(\frac{T_k + \Delta T}{T_r} \right) \cdot \frac{p_r}{\varepsilon \cdot p_a - p_r}
 \end{aligned} \right.
 \end{aligned}$$

Рис. 8.7. Подпрограмма теплового расчета дизеля $Pp_ALFA_NVR.mcd$ (начало)

$$T_a \leftarrow \frac{T_k + \Delta T + \gamma \cdot T_r}{1 + \gamma}$$

$$\eta_v \leftarrow \frac{\varepsilon}{\varepsilon - 1} \cdot \frac{P_a}{P_k} \cdot \frac{T_k}{(T_k + \Delta T)} \cdot \left(1 - \frac{P_r}{\varepsilon \cdot P_a}\right)$$

$$P_c \leftarrow P_a \cdot \varepsilon^{n_1}$$

$$T_c \leftarrow T_a \cdot \varepsilon^{n_1 - 1}$$

$$\lambda \leftarrow \frac{P_z}{P_c}$$

$$\mu \leftarrow \frac{\mu_0 + \gamma}{1 + \gamma}$$

$$q_1 \leftarrow \frac{1 + \gamma \cdot r_\alpha}{1 + \gamma}$$

$$q_2 \leftarrow \frac{\gamma \cdot r_0}{1 + \gamma}$$

$$U_{лев} \leftarrow \xi_z \cdot \frac{h_u}{(1 + \gamma) \cdot M_1} + 0 \dots$$

$$+ q_1 \cdot U_{возд}(T_c - 273) \dots$$

$$+ 0 + q_2 \cdot U_{ncz}(T_c - 273) + 8.314 \cdot \lambda \cdot T_c$$

$$t \leftarrow 1700$$

$$t_z \leftarrow \text{root} \left[\begin{array}{l} U_{лев} \dots \\ + 0 - \mu \cdot \left[\begin{array}{l} r_\alpha \cdot U_{возд}(t) \dots \\ + 0 + r_0 \cdot U_{ncz}(t) \dots \\ + 0 + 8.314 \cdot (t + 273) \end{array} \right] \end{array} \right], t$$

$$T_z \leftarrow t_z + 273$$

Рис. 8.7. Подпрограмма *Pp_ALFA_NVR.mcd* (продолжение)


```


$$\eta_i \leftarrow 8.314 \cdot \frac{p_i \cdot M_1 \cdot T_k}{h_u \cdot \eta_v \cdot p_k}$$


$$g_i \leftarrow \frac{3600000}{h_u \cdot \eta_i}$$


$$\eta_e \leftarrow \eta_i \cdot \eta_M$$


$$p_e \leftarrow p_i \cdot \eta_M$$


$$g_e \leftarrow \frac{3600000}{h_u \cdot \eta_e}$$


$$V_h \leftarrow \pi \cdot \frac{D^2}{4} (D \cdot i_{sd})$$


$$N_e \leftarrow V_h \cdot \frac{(p_e \cdot n_{\partial\theta} \cdot i_u)}{120}$$


$$S \leftarrow D \cdot i_{sd}$$

(  $\alpha$   $p_k$   $p_0$   $T_0$   $\eta_e$   $g_e$   $p_e$   $T_z$   $h_u$   $\eta_i$   $p_i$   $D$   $S$   $N_e$   $V_h$  )

Z := | m ← 0
      | for s ∈ 1, 1.2..2.6
      |   | Dm ← NeVARALFA(s)
      |   | Am ← (Dm)0,0
      |   | Bm ← (Dm)0,4
      |   | Cm ← (Dm)0,7
      |   | Em ← (Dm)0,11
      |   | Fm ← (Dm)0,13

```

Рис. 8.7. Подпрограмма *Pp_ALFA_NVR.mcd* (продолжение)

```

      |
      |  $G_m \leftarrow (D_m)_{0,5}$ 
      |  $m \leftarrow m + 1$ 
      |  $W^{(0)} \leftarrow A$ 
      |  $W^{(1)} \leftarrow B$ 
      |  $W^{(2)} \leftarrow C$ 
      |  $W^{(3)} \leftarrow E$ 
      |  $W^{(4)} \leftarrow F$ 
      |  $W^{(5)} \leftarrow G$ 
      |  $W$ 

```

$Mout2\alpha := Z$

Формирование векторов выходных параметров:

КПД эффективный $\eta E := Mout2\alpha^{(1)}$;

коэффициент избытка воздуха $Alfa := Mout2\alpha^{(0)}$;

удельный эффективный расход топлива $gE := Mout2\alpha$

температура Tz $TZ := Mout2\alpha^{(2)}$;

диаметр поршня D $DP := Mout2\alpha^{(3)}$;

эффективная мощность Ne $NE := Mout2\alpha^{(4)}$

$\eta E(a) := interp(lspline(Alfa, \eta E), Alfa, \eta E, a)$

$gE(a) := interp(lspline(Alfa, gE), Alfa, gE, a)$

$Tz(a) := interp(lspline(Alfa, TZ), Alfa, TZ, a)$

$Dn(a) := interp(lspline(Alfa, DP), Alfa, DP, a)$

$NE(a) := interp(lspline(Alfa, NE), Alfa, NE, a)$

Рис. 8.7. Подпрограмма *Pp_ALFA_NVR.mcd* (окончание)

ПРОГРАММА

теплового расчета дизеля $N=VAR$

Считывание данных матриц

$M_1 := READPRN("C:\Мои документы\TPDz\MTR_HCDz.prn")$

$V_f := READPRN("C:\Мои документы\TPDz\MTR_fuel.prn")$

☒ *Reference:C:\Мои документы\TPDz\Рр_fuile.mcd*

☒ *Reference:C:\Мои документы\TPDz\Рр_ALFA_NVR.mcd*

$$Mout2\alpha = \begin{pmatrix} 1 & 0.351 & 2.655 \times 10^3 & 1.04 & 63.351 & 229.61 \\ 1.2 & 0.364 & 2.447 \times 10^3 & 1.04 & 54.84 & 221.037 \\ 1.4 & 0.374 & 2.288 \times 10^3 & 1.04 & 48.303 & 215.103 \\ 1.6 & 0.382 & 2.162 \times 10^3 & 1.04 & 43.124 & 210.817 \\ 1.8 & 0.388 & 2.06 \times 10^3 & 1.04 & 38.921 & 207.629 \\ 2 & 0.392 & 1.976 \times 10^3 & 1.04 & 35.442 & 205.209 \\ 2.2 & 0.396 & 1.905 \times 10^3 & 1.04 & 32.515 & 203.348 \\ 2.4 & 0.399 & 1.845 \times 10^3 & 1.04 & 30.018 & 201.906 \\ 2.6 & 0.401 & 1.793 \times 10^3 & 1.04 & 27.863 & 200.788 \end{pmatrix}$$

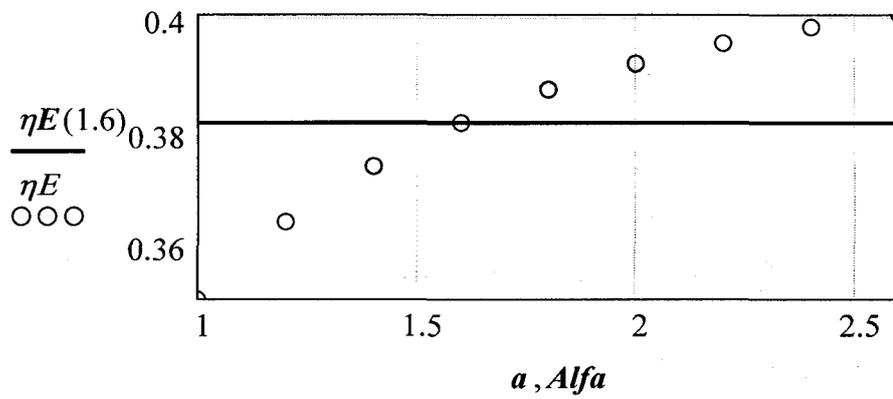
Формат матрицы $(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha)$;

результат теплового расчета дизеля $MTR_ALFA_NeVAR := Mout2\alpha$

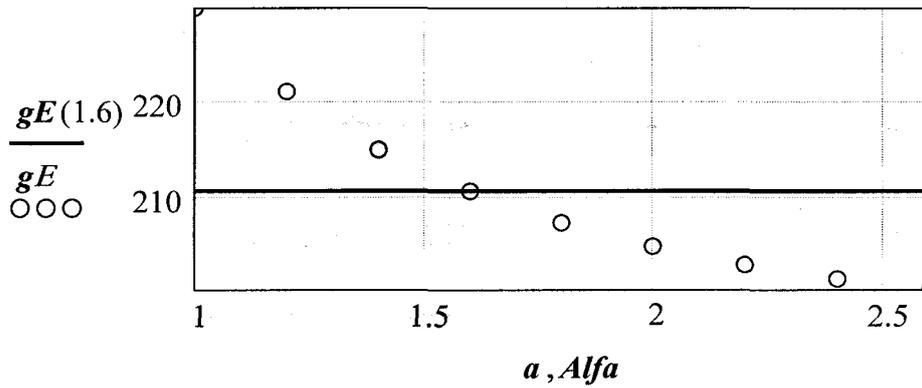
Запись результатов расчета в структурированный файл:

$WRITEPRN("MTR_ALFA_NeVAR.prn") := MTR_ALFA_NeVAR$.

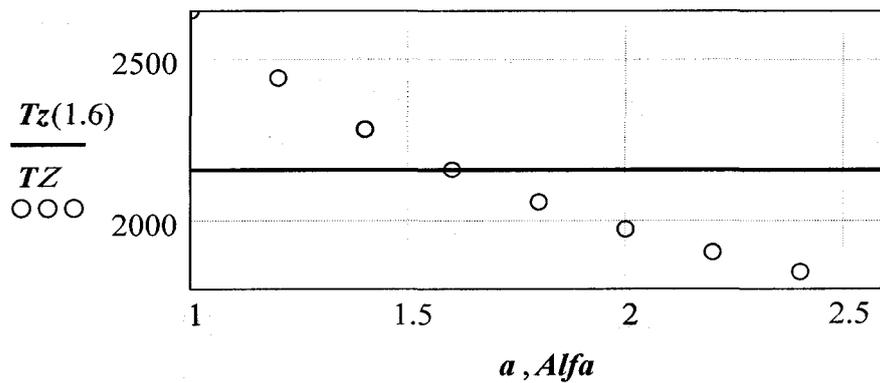
Рис. 8.8. Программа теплового расчета дизеля $R_ALFA_NVR.mcd$ (начало)



Изменение эффективного КПД в зависимости от коэффициента избытка воздуха

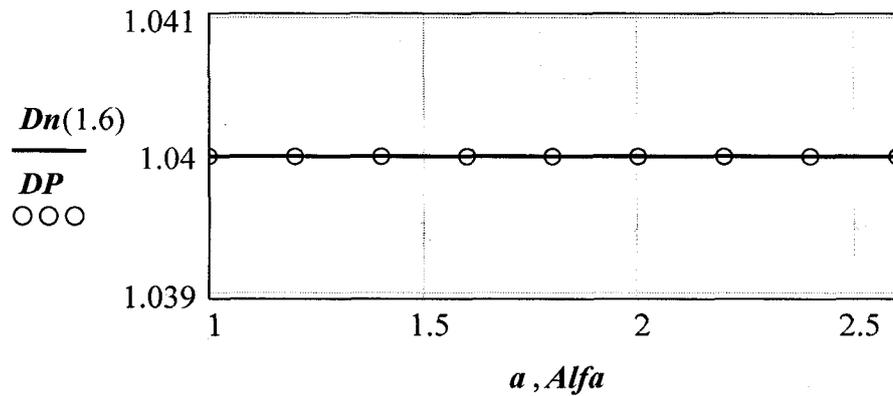


Изменение эффективного расхода топлива в зависимости коэффициента избытка воздуха

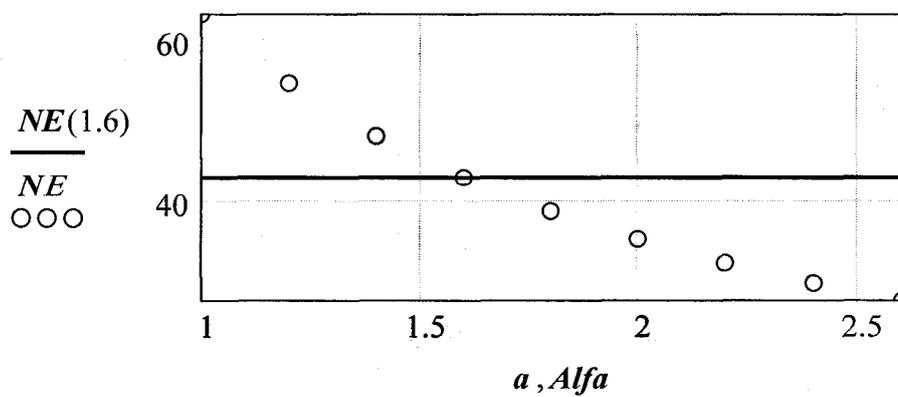


Изменение максимальной температуры в зависимости от коэффициента избытка воздуха

Рис. 8.8. Программа *R_ALFA_NVR.mcd* (продолжение)



Изменение диаметра поршня дизеля в зависимости от коэффициента избытка воздуха



Изменение мощности дизеля в зависимости от коэффициента избытка воздуха

Рис. 8.8. Программа *R_ALFA_NVR.mcd* (окончание)

ТЕМА 9. ПРОГРАММНЫЕ МОДУЛИ ВЫСОТЫ

В данном разделе приведены тексты программных модулей, предназначенных для расчета параметров цикла дизеля при изменении высоты над уровнем моря (условия высокогорья) при решении прямой и обратной задач проектирования и исследования параметров дизеля.

Блок высокогорья включает подпрограммы и программы *Pp_Hm.mcd*, *Pp_Hm_NVR*, *Protokol_Hm.mcd* (рис. 9.1 – рис. 9.3). Общая структура программного блока по своей сути аналогична рассмотренным в предыдущих разделах.

ПОДПРОГРАММА

теплового расчета дизеля $N=\text{const}$ при изменении высоты над уровнем моря (условия высокогорья)

Считывание матрицы исходных данных

$$M_1 := MTR_HCDz$$

Определение переменных:

$$\alpha := M_{1,0,0} \quad p_k := M_{1,0,1} \quad p_x := M_{1,0,2} \quad \Delta T_x := M_{1,0,3} \quad \eta_{ka} := M_{1,0,4}$$

$$p_\theta := M_{1,0,5} \quad T_\theta := M_{1,0,6} \quad k := M_{1,0,7} \quad \xi_{int} := M_{1,0,8} \quad \xi_{ex} := M_{1,1,0}$$

$$p_{kr} := M_{1,1,1} \quad \Delta T := M_{1,1,2} \quad T_r := M_{1,1,3} \quad \epsilon := M_{1,1,4} \quad n_1 := M_{1,1,5}$$

$$p_z := M_{1,1,6} \quad \xi_z := M_{1,1,7} \quad \xi_b := M_{1,1,8} \quad \phi_n := M_{1,2,0} \quad \eta_m := M_{1,2,1}$$

$$N_e := M_{1,2,2} \quad n_{\partial\theta} := M_{1,2,3} \quad i_u := M_{1,2,4} \quad i_{sd} := M_{1,2,5}$$

$$HCDHatm(H) := \left| \begin{array}{l} h_u \leftarrow H_u(V_f) \\ l_\theta \leftarrow L_\theta(V_f) \\ m_\theta \leftarrow M_\theta(V_f) \\ M_1 \leftarrow \alpha \cdot L_\theta(V_f) \\ M_2 \leftarrow M_\theta(V_f) - (\alpha - 1) \cdot L_\theta(V_f) \\ r_\theta \leftarrow \frac{M_\theta(V_f)}{[M_\theta(V_f) - (\alpha - 1) \cdot L_\theta(V_f)]} \\ r_\alpha \leftarrow \frac{[(\alpha - 1) \cdot L_\theta(V_f)]}{M_2} \\ \mu_\theta \leftarrow \frac{M_2}{M_1} \\ p'_k \leftarrow p_k \cdot \left(1 - \frac{H}{44300}\right)^{5.256} - p_x \end{array} \right.$$

Рис. 9.1. Подпрограмма влияния высоты над уровнем моря (высокогорья) на параметры дизеля $Pp_Hm.mcd$ (начало)

$$\begin{aligned}
T_k &\leftarrow (T_\theta - 0.0065 \cdot H) \cdot \left[1 - \frac{\left[\frac{p'_k}{p_\theta \cdot \left(1 - \frac{H}{44300} \right)^{5.256}} \right]^{\frac{k-1}{k}} - 1}{\eta_{ka}} \right] \cdot \Delta T_x \\
p_a &\leftarrow \xi_{int} \cdot p_k \\
p_r &\leftarrow \xi_{ex} \cdot \frac{p_k}{p_{kr}} \\
\gamma &\leftarrow \left(\frac{T_k - \Delta T}{T_r} \right) \cdot \frac{p_r}{\varepsilon \cdot p_a - p_r} \\
T_a &\leftarrow \frac{T_k - \Delta T - \gamma \cdot T_r}{1 - \gamma} \\
\eta_v &\leftarrow \frac{\varepsilon}{\varepsilon - 1} \cdot \frac{p_a}{p_k} \cdot \frac{T_k}{(T_k - \Delta T)} \cdot \left(1 - \frac{p_r}{\varepsilon \cdot p_a} \right) \\
p_c &\leftarrow p_a \cdot \varepsilon^{n_1} \\
T_c &\leftarrow T_a \cdot \varepsilon^{n_1 - 1} \\
\lambda &\leftarrow \frac{p_z}{p_c} \\
\mu &\leftarrow \frac{\mu_\theta - \gamma}{1 - \gamma} \\
q_1 &\leftarrow \frac{1 - \gamma \cdot r_\alpha}{1 - \gamma}
\end{aligned}$$

Рис. 9.1. Подпрограмма *Pp_Hm.mcd* (продолжение)

$$q_2 \leftarrow \frac{\gamma \cdot r_0}{1 + \gamma}$$

$$U_{лев} \leftarrow \xi_z \cdot \frac{h_u}{(1 + \gamma) \cdot M_1} + q_1 \cdot U_{6030}(T_c - 273)$$

$$U_{лев} \leftarrow U_{лев} + (q_2 \cdot U_{nc2}(T_c - 273) + 8.314 \cdot \lambda \cdot T_c)$$

$$t \leftarrow 1700$$

$$t_z \leftarrow \text{root} \left[\left[U_{лев} - \mu \cdot \begin{bmatrix} r_\alpha \cdot U_{6030}(t) \dots \\ + 0 + r_\theta \cdot U_{nc2}(t) \dots \\ + 0 + 8.314 \cdot (t + 273) \end{bmatrix}, t \right] \right]$$

$$T_z \leftarrow t_z + 273$$

$$\rho \leftarrow \mu \cdot \frac{T_z}{\lambda \cdot T_c}$$

$$\delta \leftarrow \frac{\varepsilon}{\rho}$$

$$U'' \leftarrow (\xi_b - \xi_z) \cdot \frac{h_u}{(\mu_\theta + \gamma) \cdot M_1}$$

$$n \leftarrow 1.15$$

$$n.2 \leftarrow \text{root} \left[\left[U'' - \begin{bmatrix} 8.314 \cdot \frac{(T_z - T_z \cdot \delta^{1-n})}{n-1} - 0 \dots \\ + r_\alpha \cdot (U_{6030}(t_z) - U_{6030}(T_z \cdot \delta^{1-n} - 273)) - 0 \dots \\ + r_\theta \cdot (U_{nc2}(t_z) - U_{nc2}(T_z \cdot \delta^{1-n} - 273)) \end{bmatrix}, n \right] \right]$$

$$p.b \leftarrow p.z \cdot \delta^{-n.2}$$

$$T.b \leftarrow T.z \cdot \delta^{1-n.2}$$

$$p.ip \leftarrow \frac{p.c}{\varepsilon - 1} \cdot \begin{bmatrix} \lambda \cdot (\rho - 1) + \lambda \cdot \frac{\rho}{n.2 - 1} \cdot \left(1 - \frac{T.b}{T.z} \right) \dots \\ + 0 - \frac{1}{n.1 - 1} \cdot \left(1 - \frac{T.a}{T.c} \right) \end{bmatrix}$$

Рис. 9.1. Подпрограмма *Pp_Hm.mcd* (продолжение)

```

p.i ←  $\phi.n \cdot p.ip$ 
η.i ←  $8.314 \cdot \frac{p.i \cdot M.1 \cdot T.k}{h.u \cdot \eta.v \cdot p.k}$ 
g.i ←  $\frac{3600000}{h.u \cdot \eta.i}$ 
η.e ←  $\eta.i \cdot \eta.m$ 
p.e ←  $p.i \cdot \eta.m$ 
g.e ←  $\frac{3600000}{h.u \cdot \eta.e}$ 
V.h ←  $120 \cdot \frac{N.e}{p.e \cdot n.\delta\delta \cdot i.u}$ 
D ←  $\sqrt[3]{\frac{V.h}{0.785 \cdot i.sd}}$ 
S ←  $D \cdot i.sd$ 
(η.e g.e p.e T.z h.u η.i p.i D S V.h α N.e H)

```

Формирование матрицы результатов расчета в цикле[^]

```

MoutHatm :=  $\left| \begin{array}{l} H \leftarrow 0 \\ Vout \leftarrow HCDHatm(H) \\ \text{for } \epsilon \in 1,2..35 \\ \quad Vout \leftarrow stack(Vout, HCDHatm(H + \epsilon \cdot 100)) \\ Vout \end{array} \right.$ 

```

Формат матрицы результатов

(*η_e g_e p_e T_z h_u η_i p_i D S V_h α N_e H*)

Формирование векторов данных

$\eta E := MoutHatm^{(0)}$ $H := MoutHatm^{(12)}$ $gE := MoutHatm^{(1)}$
 $TZ := MoutHatm^{(3)}$ $DP := MoutHatm^{(7)}$ $NE := MoutHatm^{(11)}$

Рис. 9.1. Подпрограмма *Pp_Hm.mcd* (продолжение)

Интерполяция результатов расчета кубическим сплайном:

$\eta E(h) := \text{interp}(\text{lspline}(H, \eta E), H, \eta E, h)$

$gE(h) := \text{interp}(\text{lspline}(H, gE), H, gE, h)$

$Tz(h) := \text{interp}(\text{lspline}(H, TZ), H, TZ, h)$

$Dn(h) := \text{interp}(\text{lspline}(H, DP), H, DP, h)$

$NE(h) := \text{interp}(\text{lspline}(H, NE), H, NE, h)$

Конец подпрограммы *Pp_Hm.mcd*

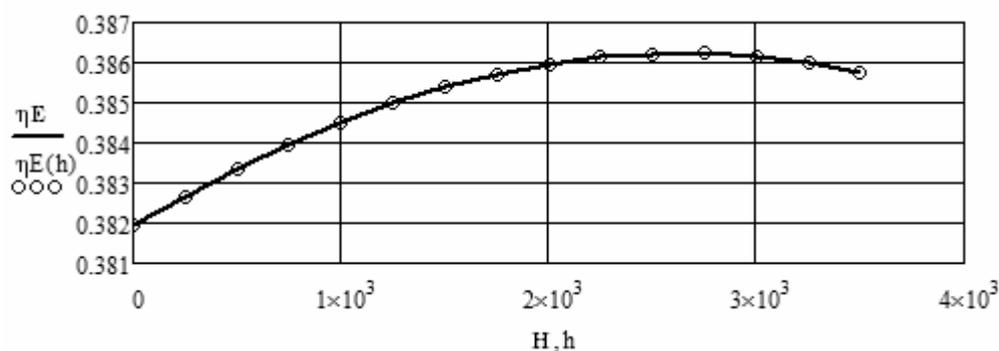
Рис. 9.1. Подпрограмма *Pp_Hm.mcd* (окончание)

ПРОГРАММА

теплового расчета дизеля $N=\text{const}$ при изменении высоты над уровнем моря (условия высокогорья)

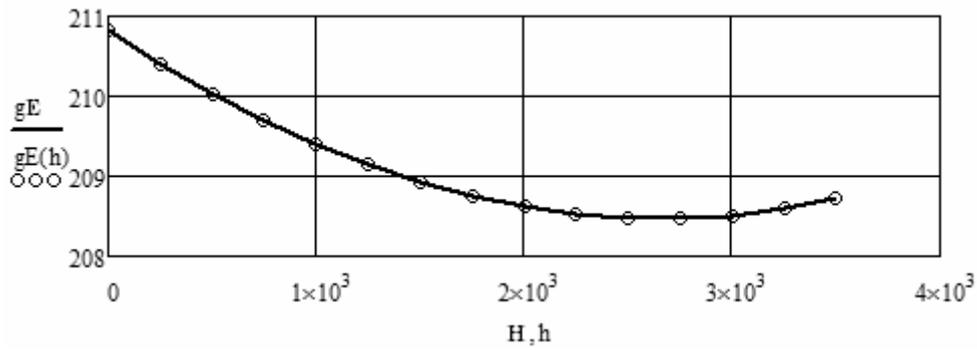
Фрагмент матрицы результатов теплового расчета дизеля:

	0	1	2	3	4
0	0.382	210.817	0.832	$2.162 \cdot 10^3$	$4.471 \cdot 10^4$
1	0.383	210.408	0.814	$2.169 \cdot 10^3$	$4.471 \cdot 10^4$
2	0.383	210.035	0.795	$2.177 \cdot 10^3$	$4.471 \cdot 10^4$
3	0.384	209.701	0.777	$2.184 \cdot 10^3$	$4.471 \cdot 10^4$
4	0.385	209.404	0.759	$2.192 \cdot 10^3$	$4.471 \cdot 10^4$
5	0.385	209.146	0.741	$2.201 \cdot 10^3$	$4.471 \cdot 10^4$
6	0.385	208.928	0.724	$2.209 \cdot 10^3$	$4.471 \cdot 10^4$
7	0.386	208.751	0.706	$2.218 \cdot 10^3$	$4.471 \cdot 10^4$
8	0.386	208.615	0.689	$2.227 \cdot 10^3$	$4.471 \cdot 10^4$
9	0.386	208.521	0.672	$2.237 \cdot 10^3$	$4.471 \cdot 10^4$
10	0.386	208.471	0.655	$2.247 \cdot 10^3$	$4.471 \cdot 10^4$
11	0.386	208.466	0.638	$2.257 \cdot 10^3$	$4.471 \cdot 10^4$
12	0.386	208.507	0.622	$2.268 \cdot 10^3$	$4.471 \cdot 10^4$
13	0.386	208.595	0.606	$2.279 \cdot 10^3$	$4.471 \cdot 10^4$
14	0.386	208.733	0.589	$2.291 \cdot 10^3$...

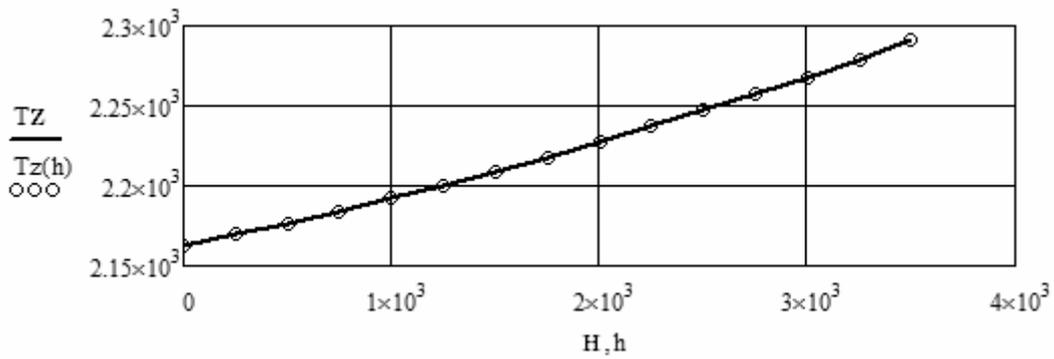


Изменение эффективного КПД дизеля в зависимости от высоты над уровнем моря

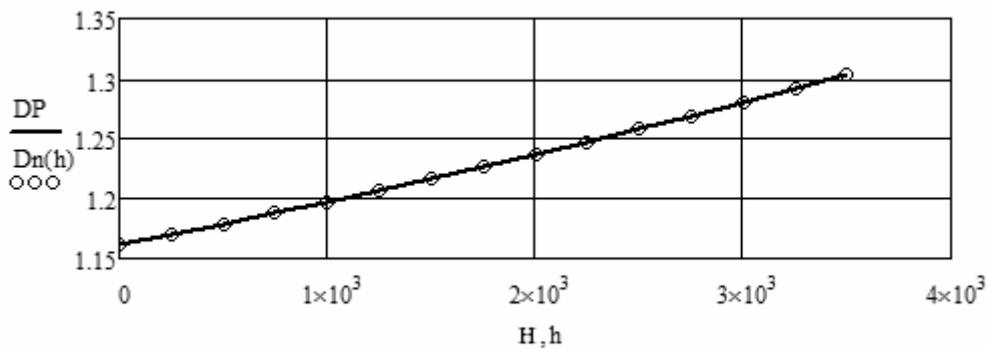
Рис. 9.2. Программа влияния высоты над уровнем моря (высокогорья) на параметры дизеля $R_{Hm.mcd}$ (начало)



Изменение удельного эффективного расхода топлива дизеля в зависимости от высоты над уровнем моря

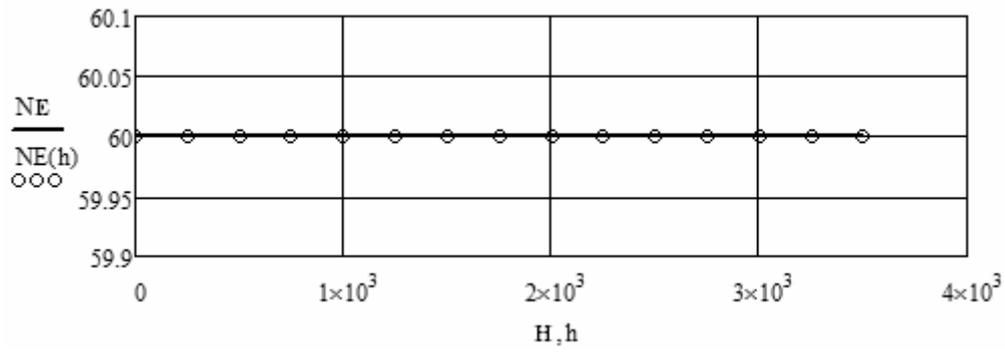


Изменение максимальной температуры дизеля в зависимости от высоты над уровнем моря



Изменение диаметра поршня дизеля в зависимости от высоты над уровнем моря

Рис. 9.2. Программа влияния высоты над уровнем моря (высокогорья) на параметры дизеля *R_Hm.mcd* (продолжение)

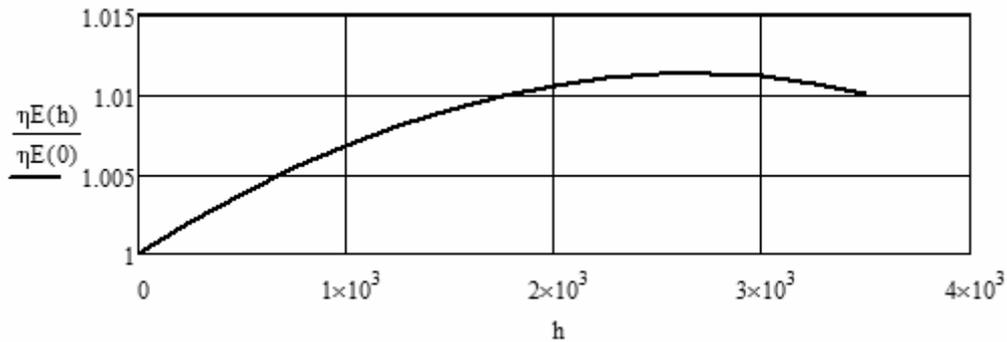


Изменение мощности дизеля в зависимости от высоты над уровнем моря

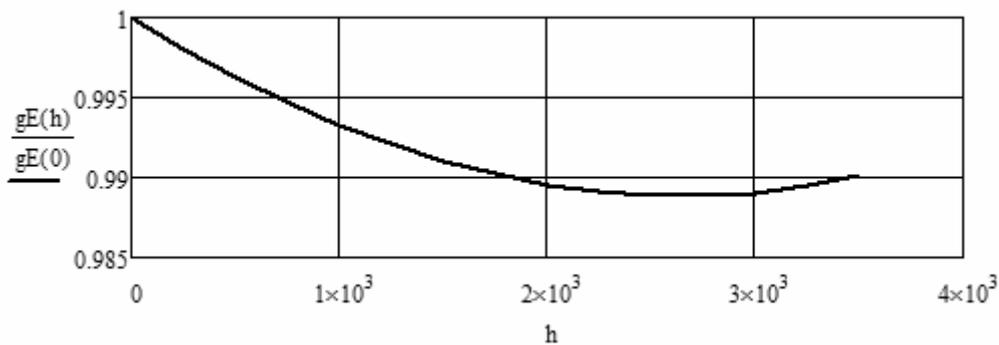
Рис. 9.2. Программа влияния высоты над уровнем моря (высокогорья) на параметры дизеля *R_Hm.mcd* (окончание)

ПРОТОКОЛ

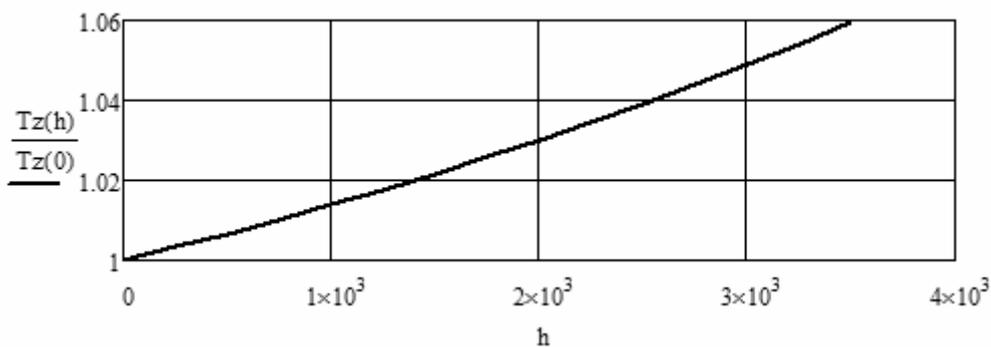
теплого расчета дизеля при изменении высоты над уровнем моря (условия высокогорья) *Protokol_Hm.mcd*



Относительное изменение эффективного КПД дизеля в зависимости от высоты над уровнем моря (прямая задача)

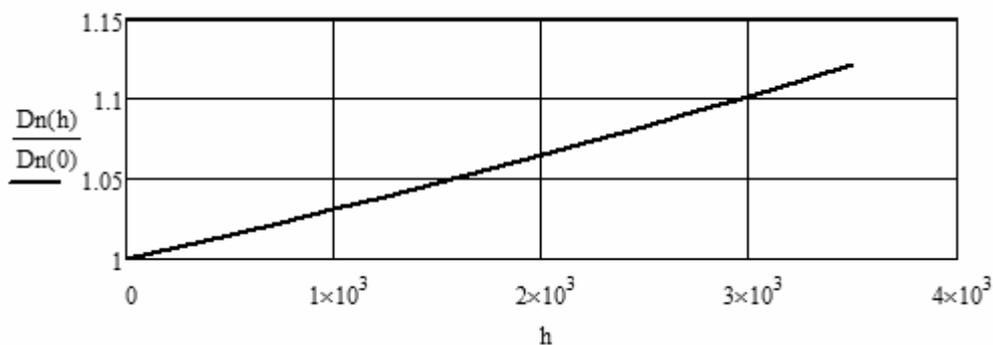


Относительное изменение эффективного расхода топлива при изменении высоты над уровнем моря (прямая задача)



Относительное изменение максимальной температуры в зависимости от высоты над уровнем моря (прямая задача)

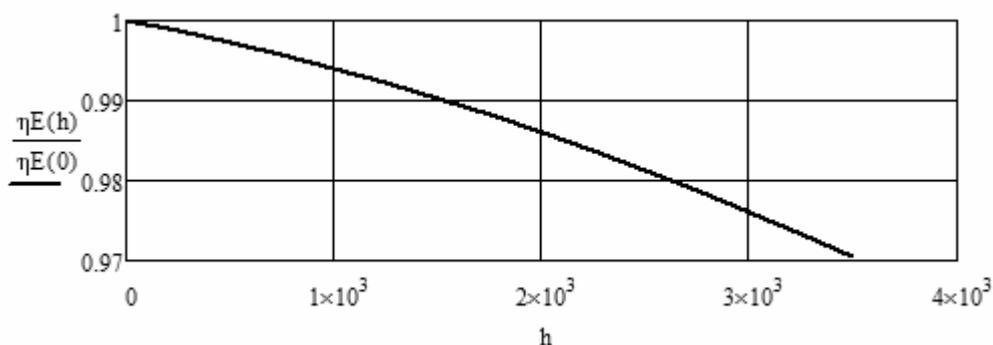
Рис. 9.3. Программа – протокол расчета влияния высоты над уровнем моря на параметры дизеля (начало)



Относительное изменение диаметра поршня двигателя в зависимости от высоты над уровнем моря (прямая задача)

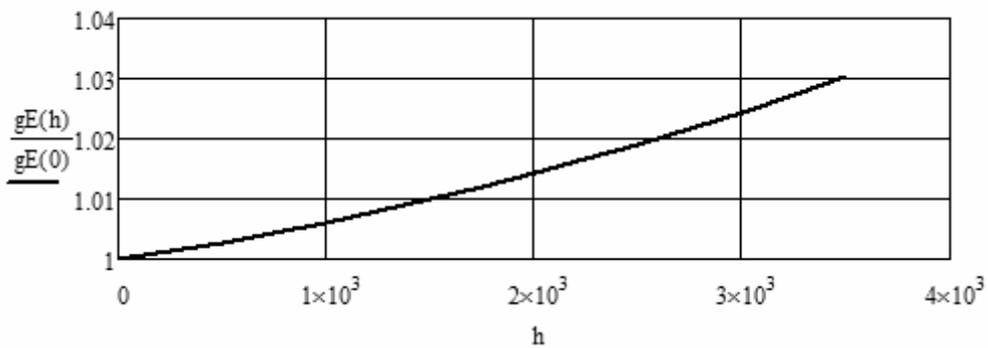


Относительное изменение мощности дизеля в зависимости от высоты над уровнем моря (прямая задача)

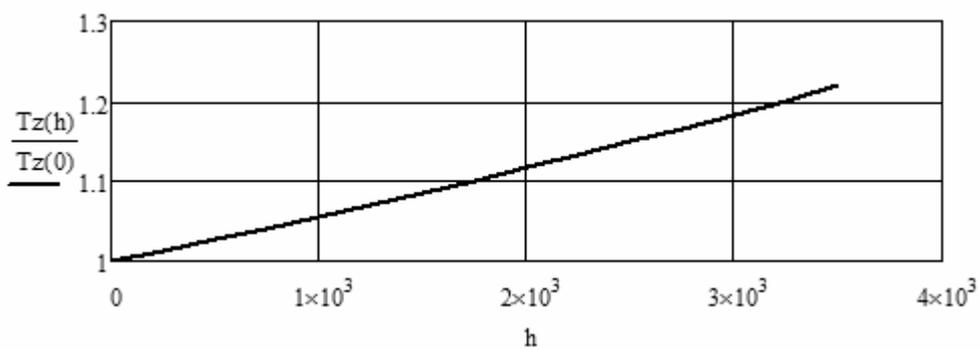


Относительное изменение эффективного КПД дизеля в зависимости от высоты над уровнем моря (обратная задача)

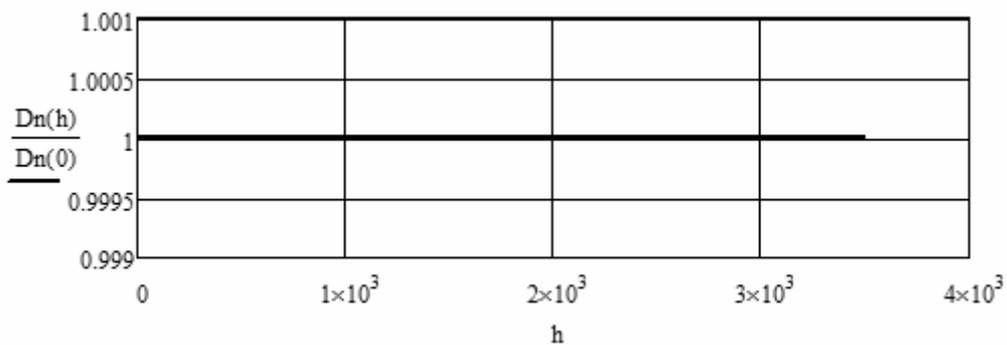
Рис. 9.3. Программа – протокол расчета влияния высоты над уровнем моря на параметры дизеля (продолжение)



Относительное изменение эффективного расхода топлива дизеля в зависимости от высоты над уровнем моря (обратная задача)

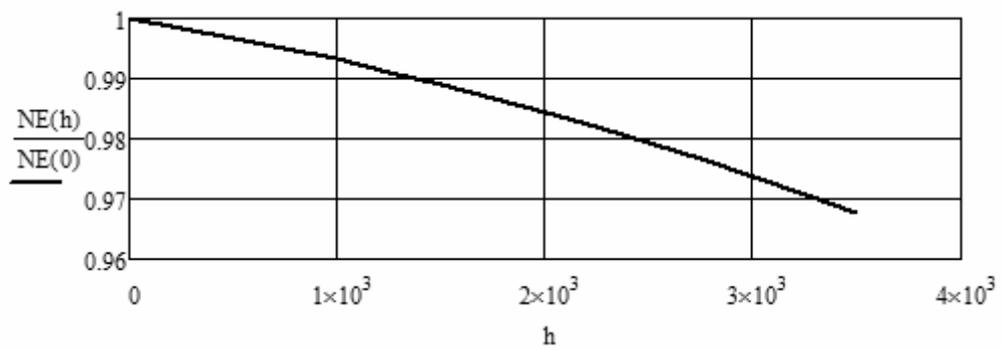


Относительное изменение максимальной температуры в зависимости от высоты над уровнем моря (обратная задача)



Относительное изменение диаметра поршня двигателя в зависимости от высоты над уровнем моря (обратная задача)

Рис. 9.3. Программа – протокол расчета влияния высоты над уровнем моря на параметры дизеля (продолжение)



Относительное изменение мощности дизеля в зависимости от высоты над уровнем моря (обратная задача)

Рис. 9.3. Программа – протокол расчета влияния высоты над уровнем моря на параметры дизеля (окончание)

ОПИСАНИЕ КУРСА И ПРОГРАММА

Цели и задача курса:

Целью курса «Повышение эффективности работы автомобильных дизелей в условиях высокогорья и жаркого климата» является изучение студентами основных аспектов работы с интегрированной системой *Mathcad*, позволяющей грамотно ориентироваться в вопросах получения, обработки, интерпретации численной информации, освоение основных принципов построения и работы с системами автоматизированного расчета для проведения однофакторного и многофакторного численного исследования и изучения основных закономерностей влияний параметров окружающей среды (состояния атмосферы) и высоты над уровнем моря на основные показатели работы автомобильного дизеля.

Для достижения указанной цели в курсе решаются следующие задачи. Изучение основных способов создания файлов и файлов на основе шаблонов. Изучение методов импорта и экспорта числовой и текстовой информации, организация гиперссылок. Выполнение вычислений с помощью встроенных функций и функций пользователя, векторов и матриц. Изучить приемы программирования и применения условных операторов и операторов цикла. Уметь применять линейную интерполяцию и кубическую сплайн-интерполяцию при обработке экспериментальных и расчетных данных. Изучить построение графиков по данным векторов, векторов и ранжированной переменной, функций. Получить навыки ввода – вывода во внешние файлы. Изучить основные принципы разработки систем автоматизированного расчета на модульном принципе. Знать этапы разработки программных модулей, способы установления информационных связей между модулями, способы сборки программ. Изучить структуру системы автоматизированного расчета показателей работы дизеля при изменении

параметров атмосферы и высоты над уровнем моря. Изучить программные модули САР: модули подготовки и ввода исходных данных, подпрограммы расчета влияния изменяющихся параметров на работу дизеля, расчетные модули системы, программы-протоколы, текстовые структурированные файлы.

Область знаний: технические науки, энергомашиностроение, тепловые двигатели, поршневые и комбинированные двигатели внутреннего сгорания.

Уровень обучения: бакалавриат.

Курс является курсом по выбору для направления «Энергомашиностроение», специальность «Поршневые и комбинированные двигатели внутреннего сгорания», а также специальности «Эксплуатация автомобильного транспорта» направления «Производство и эксплуатация транспортных средств».

Инновационность курса

по содержанию заключается в том, что в него включены последние методические разработки кафедры;

по методике преподавания, так как она основана на использовании компьютерных технологий, а именно: в формах презентации лекционного материала, в проведении лабораторных работ с использованием современных компьютерных интегрированных систем;

по литературе, так как в ней используются не только учебная литература, а результаты собственных методических разработок и разработок совместно с другими университетами.

Сведения об авторе курса: Виноградов Леонид Валерьевич, кандидат технических наук, доцент, профессор кафедры теплотехники и тепловых двигателей.

Структура курса: (количество аудиторных часов/ самостоятельная работа 72/72. Аудиторные часы (72) включают в себя 36 часов лекций, 36 часов лабораторных работ.

Кредит - 2

Темы лекций.

1. Введение – интегрированная система *Mathcad*, ее назначение, основные вычислительные возможности. Создание документа на основе шаблона. Ввод и редактирование формул. Ввод и редактирование текста. Импорт текста. Математические символы внутри текста. Гипер ссылки. Посылка документов по электронной почте. Переменные и функции. Присваивание переменным значений. Определение функции пользователя. Матричные операторы. Транспонирование. Функции создания матриц. Слияние и разбиение матриц. Решение уравнений. (4/4)

2. Язык программирования в *Mathcad*. Создание программ. Условные операторы (*if, otherwise*). Операторы цикла (*for, while*). Примеры программирования. Линейная интерполяция. Кубическая сплайн-интерполяция. (4/4)

3. Числовой ввод-вывод. Двумерные графики: XY-график двух векторов; XY-график вектора и ранжированной переменной; XY-графики функций. Примеры построения различного вида графиков. Ввод-вывод во внешние файлы. Текстовые файлы. Графические файлы. Ссылки и гиперссылки. (4/4)

4. Понятие систем автоматизированного расчета (САР). Структура и основные компоненты САР. Этапы развития и показатели качества.

Модульный принцип разработки САР. Восходящий и нисходящий методы разработки. Разработка и оформление модулей. Согласованность модулей. Сборка программ. Библиотека модулей. Практические примеры САР. (4/4)

5. Понятие прямой и обратной задачи при исследовании показателей работы дизеля при переменных условиях окружающей среды. Программа подготовки и ввода исходных данных *ID_TPD_1.mcd* (прямая задача). Программа подготовки и ввода исходных данных *ID_TPD_2.mcd* (обратная задача).

Стандартная атмосфера. Изменение параметров атмосферы (давление, температура, плотность) в зависимости от высоты над уровнем моря. Программный модуль (ПМ) *Pp_MATM.mcd*. (4/4)

6. ПМ расчета теплоты сгорания, состава продуктов сгорания и внутренней энергии воздуха и газа *Pp_FUEL.mcd*. ПМ теплового расчета параметров дизеля (прямая задача $N_e = const$) *Pp_HCD_1.mcd*. ПМ теплового расчета параметров дизеля (обратная задача $D = const, i_{sd} = const$) *Pp_HCD_2.mcd*. (4/4)

7. Блок ПМ расчета влияния изменения давления атмосферы на показатели дизеля. *Pp_P0.mcd, PP_P0_NVR.mcd, R_P0.mcd, R_P0_NVR.mcd*. Блок ПМ расчета влияния изменения температура атмосферы на показатели дизеля. *Pp_T0.mcd, PP_T0_NVR.mcd, R_T0.mcd, R_T0_NVR.mcd*.

8. Блок ПМ расчета влияния изменения давления наддува на показатели дизеля. *Pp_PK.mcd, PP_PK_NVR.mcd, R_PK.mcd, R_PK_NVR.mcd*. Блок ПМ расчета влияния изменения коэффициента избытка воздуха на показатели работы дизеля. *Pp_ALFA.mcd, PP_ALFA_NVR.mcd, R_ALFA.mcd, R_ALFA_NVR.mcd*. (4/4)

9. Блок ПМ расчета влияния изменения высоты над уровнем моря на показатели дизеля. *Pp_H.mcd, PP_H_NVR.mcd, R_H.mcd, R_H_NVR.mcd*. Блок ПМ протоколов результатов численного исследования (прямая и обратная задачи) влияния изменения параметров на показатели дизеля. *P_P0.mcd, P_ALFA.mcd, P_T0.mcd, P_PK.mcd, P_H.mcd, P_INT.mcd*.

Темы лабораторных работ (36/36).

1. Провести численное исследование влияния изменения начального атмосферного давления на показатели работы вновь проектируемого дизеля. Собрать программу из *template* – программ. Построить графики. Интегральную матрицу расчетов вывести во внешний файл. (6/6)

2. Провести численное исследование влияния изменения начальной температуры окружающей среды на показатели работы вновь проектируемого дизеля. Собрать программу из *template* – программ. Построить графики. Интегральную матрицу расчетов вывести во внешний файл. (6/6)

3. Провести численное исследование влияния коэффициента избытка воздуха на показатели работы дизеля. Собрать программу из *template* – программ. Построить графики. Интегральную матрицу расчетов вывести во внешний файл. (6/6)

4. Провести численное исследование влияния изменения давления наддува на показатели работы вновь спроектированного дизеля. Собрать программу из *template* – программ. Построить графики. Интегральную матрицу расчетов вывести во внешний файл. (6/6)

5. Провести численное исследование влияния изменения высоты над уровнем моря (стандартная атмосфера) на показатели работы существующего дизеля. Собрать программу из *template* – программ. Построить графики. Интегральную матрицу расчетов вывести во внешний файл. (6/6)

6. Провести численное исследование влияния изменения высоты над уровнем моря (стандартная атмосфера) на показатели работы вновь проектируемого дизеля. Собрать программу из *template* – программ. Построить графики. Интегральную матрицу расчетов вывести во внешний файл. (6/6)

Описание системы контроля знаний.

Текущая аттестация студентов – рубежный контроль производится в плановые часы занятий. Каждый студент индивидуально выполняет пись-

менную работу – тест. Рубежный контроль проводится два раза в семестр (в октябре, ноябре осеннего семестра и марте, апреле весеннего семестра) с проставлением результатов с 20-го по последний день текущего месяца на сайте контроля текущей успеваемости студентов. После наступления нового месяца, проставленные баллы за предыдущий месяц, не могут быть исправлены. Баллы проставляются нарастающим итогом и в конце курса по их количеству определяется дальнейшее действия студента: согласие на автоматическое выставление оценки или участие в экзамене.

Общие правила выполнения контрольных заданий.

Устанавливаются и доводятся до сведения студентов следующие условия начисления баллов по курсу и на рубежной аттестации:

- посещение лекций и лабораторных работ за 2 часа (пару) – 1 балл (всего максимум 18 - лекции, 18 - лабораторные дают в сумме 36 баллов);
- защита лабораторных работ за 1 защищенную работу до 6 баллов (всего максимум студент за защиту 6-х лабораторных работ может получить в сумме 36 балла);
- рубежный контроль, каждый до 8 баллов (всего максимум студент на 2-х аттестациях может получить в сумме 16 балла)

Студент за время обучения в семестре может набрать до 100 баллов.

Пример

Шкала оценок, итоговые оценки (методика выставления).

Сумма баллов, набранная студентом на аттестациях, позволяет получить оценку по итогам работы в семестре и не сдавать экзаменов, исходя из следующей шкалы (1-й и 2-й столбцы таблицы 1). Студенты, получившие оценки за экзамены по результатам работы в семестре, но претендующие на получение более высокой оценки, могут участвовать в сдаче экзаменов в сессию (см. 3, 4 и 5 столбцы таблицы 1).

Таблица 1

Сумма баллов, полученная за семестр на аттестациях	Автоматическая оценка за работу в семестре	Баллы, полученные на экзамене	Общая сумма баллов за работу в семестре и на экзамене	Итоговая оценка
91 – 100	5	-	91 - 100	5
76 – 90	4	0 -25	76 – 100 более 100	4 5
56 – 75	3	0 – 25	56 – 85 86 - 100	3 4
35 – 55	нет, обязательная сдача экзамена	0 - 25	60 - 80	3
менее 35	нет	недопуск	менее 35	2

Таблица 2

Сумма баллов, полученная за семестр на аттестациях	Автоматическая оценка за работу в семестре	Баллы, полученные на зачете	Общая сумма баллов за работу в семестре и на зачете	Итоговая оценка
> 55	Зачет	-	> 55	Зачет
35 - 55	нет, обязательная сдача зачета	0 - 25	не менее 60	Зачет
менее 35	нет	-	менее 35	Незачет

Зачет можно получить по результатам работы в семестре (см. 1-й и 2-й столбцы таблицы 2) или приняв участие в зачетной сессии (см. 3, 4 и 5 столбцы таблицы 2).

Результаты проставляются в зачетные ведомости и книжки:

- зачеты – в день проведения зачета;
- экзамены – в день проведения экзамена при наличии штампа допуска к сессии.

Программа курса УМК.

Аннотированное содержание курса УМК

Лекционный курс имеет трудоемкость 1 кредит.

Интегрированная система *Mathcad*, ее назначение и возможности. Создание документа на основе шаблона. Ввод и редактирование формул. Ввод и редактирование текста. Импорт текста. Математические символы внутри текста. Гиперссылки. Посылка документов по электронной почте. Переменные и функции. Присваивание переменным значений. Определение функции пользователя. Матричные операторы. Транспонирование. Функции создания матриц. Слияние и разбиение матриц. Решение уравнений. Язык программирования в *Mathcad*. Создание программ. Условные операторы (*if, otherwise*). Операторы цикла (*for, while*). Примеры программирования. Линейная интерполяция. Кубическая сплайн-интерполяция. Числовой ввод-вывод. Двумерные графики: XY-график двух векторов; XY-график вектора и ранжированной переменной; XY-графики функций. Ввод-вывод во внешние файлы. Текстовые файлы. Графические файлы. Ссылки и гиперссылки. Понятие систем автоматизированного расчета (САР). Структура и основные компоненты САР. Этапы развития и показатели качества. Модульный принцип разработки САР. Восходящий и нисходящий методы разработки. Разработка и оформление модулей. Согласованность модулей. Сборка программ. Библиотека модулей. Практические примеры САР. Прямая и обратная задачи исследования показателей работы дизеля. Программа подготовки и ввода исходных данных для расчета показателей работы дизеля в изменяющихся условиях окружающей среды *ID_ТРД_1.mcd* (прямая задача). Программа подготовки и ввода исходных данных *ID_ТРД_2.mcd* (обратная задача). Стандартная атмосфера. Изменение параметров атмосферы (давление, температура, плотность) в зависимости от высоты над уровнем моря. Программный модуль (ПМ) *Pp_MATM.mcd*. ПМ расчета теплоты сгорания, состава продуктов сгорания

и внутренней энергии воздуха и газа $Pp_FUEL.mcd$. ПМ теплового расчета параметров дизеля (прямая задача $N_e=const$) $Pp_HCD_1.mcd$. ПМ теплового расчета параметров дизеля (обратная задача $D = const, i_{sd} = const$) $Pp_HCD_2.mcd$. Блок ПМ расчета влияния изменения давления атмосферы на показатели дизеля. $Pp_P0.mcd, PP_P0_NVR.mcd, R_P0.mcd, R_P0_NVR.mcd$. Блок ПМ расчета влияния изменения температура атмосферы на показатели дизеля. $Pp_T0.mcd, PP_T0_NVR.mcd, R_T0.mcd, R_T0_NVR.mcd$. Блок ПМ расчета влияния изменения давления наддува на показатели дизеля. $Pp_PK.mcd, PP_PK_NVR.mcd, R_PK.mcd, R_PK_NVR.mcd$. Блок ПМ расчета влияния изменения коэффициента избытка воздуха на показатели работы дизеля. $Pp_ALFA.mcd, PP_ALFA_NVR.mcd, R_ALFA.mcd, R_ALFA_NVR.mcd$. Блок ПМ расчета влияния изменения высоты над уровнем моря на показатели дизеля. $Pp_H.mcd, PP_H_NVR.mcd, R_H.mcd, R_H_NVR.mcd$. Блок ПМ протоколов результатов численного исследования (прямая и обратная задачи) влияния изменения параметров на показатели дизеля. $P_P0.mcd, P_ALFA.mcd, P_T0.mcd, P_PK.mcd, P_H.mcd, P_INT.mcd$.

Лабораторные занятия имеют общую трудоемкость 1 кредит.

Лабораторная работа 1. (6 часов). Провести численное исследование влияния изменения начального атмосферного давления на показатели работы вновь проектируемого дизеля.

Лабораторная работа 2. (6 часов). Провести численное исследование влияния изменения начальной температуры окружающей среды на показатели работы вновь проектируемого дизеля.

Лабораторная работа 3. (6 часов). Провести численное исследование влияния коэффициента избытка воздуха на показатели работы дизеля.

Лабораторная работа 4. (6 часов). Провести численное исследование

влияния изменения давления наддува на показатели работы вновь спроектированного дизеля.

Лабораторная работа 5. (6 часов). Провести численное исследование влияния изменения высоты над уровнем моря (стандартная атмосфера) на показатели работы существующего дизеля.

Лабораторная работа 6. (6 часов). Провести численное исследование влияния изменения высоты над уровнем моря (стандартная атмосфера) на показатели работы вновь проектируемого дизеля.

ПРИМЕР:

выполнения лабораторной работы «Численное исследование влияния коэффициента избытка воздуха на показатели работы дизеля» (прямая задача)

Система автоматизированного расчета включает четыре программных модулей библиотечного типа с расширением *.mcd*:

ID_HCDz.mcd – программа подготовки и ввода исходных данных (текст программы приведен ниже;

Pp_FUEL.mcd – подпрограмма расчета характеристик топлива и продуктов сгорания;

Pp_HCD_ALFA.mcd – подпрограмма теплового расчета показателей дизеля при изменении коэффициента избытка воздуха;

R_Pp_HCD_ALFA.mcd – программа расчета показателей работы дизеля.

ПОДПРОГРАММА

ввода исходных данных для теплового расчета дизеля

$ID_HCDz.mcd$

$N_e := 60$ - эффективная мощность дизеля, кВт;

$n_{\partial\theta} := 1600$ - число оборотов двигателя, об/мин;

$i_u := 4$ - число цилиндров двигателя;

$i_{sd} := 1.1$ - относительный ход поршня;

$\alpha := 1.6$ - коэффициент избытка воздуха;

$p_0 := 0.103$ - давление во впускном коллекторе двигателя, МПа;

$T_0 := 293$ - температура воздуха во впускном коллекторе двигателя,

К;

$p_k := 0.18$ - давление наддува, МПа;

$p_x := 0.003$ - потеря давления в охладителе надувочного воздуха (ОНВ), МПа;

$\Delta T_x := 20$ - понижение температуры воздуха в ОНВ, град.;

$k := 1.4$ - показатель адиабаты воздуха;

$\xi_{int} := 0.95$ - коэффициент гидравлических потерь на впуске для дизелей;

$p_{kr} := 1.15$ - отношение давления после компрессора к давлению перед турбиной;

$\Delta T := 10$ - подогрев заряда, град;

$T_r := 900$ - температура остаточных газов, К;

$\varepsilon := 14$ - степень сжатия;

$p_z := 8$ - максимальное давление сгорания, МПа;

$\xi_z := 0.8$ - коэффициент;

$\xi_b := 0.9$ - коэффициент;

$\phi_n := 0.93$ - коэффициент полноты диаграммы для дизеля;

$\eta_{ka} := 0.72$ - КПД адиабатический компрессора;

$\xi_{ex} := 1.2$ - коэффициент;

$n_l := 1.36$ - показатель политропы;

$\eta_m := 0.87$ - механический КПД;

$D := 1.04$ - диаметр цилиндра, дм;

$S := D \cdot i_{sd}$ - ход поршня, дм;

$$V_h := \pi \cdot \frac{D^2}{4} (D \cdot i_{sd}) \quad \blacksquare \quad \text{- рабочий объем цилиндра двигателя, дм}^3;$$

Состав топлива:

$C := 0.87$ - массовая доля углерода в топливе;

$H := 0.121$ - массовая доля водорода в топливе;

$Se := 0.005$ - массовая доля серы в топливе;

$O := 0.004$ - массовая доля кислорода в топливе;

$W := 0$ - массовая доля воды в топливе.

Формируем матрицу исходных данных

$$MID_HCDz := \begin{pmatrix} N_e & n_{\partial v} & i_{\psi} & i_{sd} & \alpha & p_0 \\ T_0 & p_k & p_x & \Delta T_x & k & \xi_{int} \\ p_{kr} & \Delta T & T_r & \varepsilon & p_z & \xi_z \\ \xi_b & \phi_n & \eta_{ka} & \xi_{ex} & n_l & \eta_m \\ 0 & 0 & 0 & C & H & Se \\ O & W & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$MID_HCDz = \begin{pmatrix} 60 & 1.6 \times 10^3 & 4 & 1.1 & 1.6 & 0.103 \\ 293 & 0.18 & 3 \times 10^{-3} & 20 & 1.4 & 0.95 \\ 1.15 & 10 & 900 & 14 & 8 & 0.8 \\ 0.9 & 0.93 & 0.72 & 1.2 & 1.36 & 0.87 \\ 0 & 0 & 0 & 0.87 & 0.121 & 5 \times 10^{-3} \\ 4 \times 10^{-3} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Записываем матрицу исходных данных в текстовой структурированный файл

`WRITEPRN("MID_HCDz.prn") := MID_HCDz`

Записываем матрицу состава топлива

`MID_FUEL := (C H Se O W)`

`WRITEPRN("MID_FUEL.prn") := MID_FUEL`

Конец программы `ID_HCDz`

ПРОГРАММА

Расчета характеристик топлива и его продуктов сгорания для теплового расчета двигателя

Низшая теплота сгорания топлива рассчитывается по формуле Д.И. Менделеева

$$V_f := (C \ H \ Se \ O \ W)$$

$$V_f := V_f^T$$

$$H_u(V_f) := \left| \begin{array}{l} h \leftarrow V_{f0} \cdot 33910 \\ h \leftarrow h + V_{f1} \cdot 125600 \\ h \leftarrow h + (V_{f2} - V_{f3}) \cdot 10890 \\ h \leftarrow h + (9 \cdot V_{f1} - V_{f4}) \\ h \end{array} \right.$$
$$h_u := H_u(V_f)$$

Теоретически необходимое количество воздуха для сгорания 1 кг топлива

$$L_0(V_f) := \left| \begin{array}{l} s \leftarrow \frac{V_{f0}}{12} \\ s \leftarrow s + \frac{V_{f1}}{4} \\ s \leftarrow s + \frac{(V_{f2} - V_{f3})}{32} \\ s \leftarrow \frac{s}{0.21} \\ s \end{array} \right.$$

$$l_0 := L_0(V_f)$$

Количество продуктов сгорания 1 кг топлива $\alpha = 1$

$$M_0(V_f) := \begin{cases} s \leftarrow \frac{V_{f0}}{12} \\ s \leftarrow s + \frac{V_{f1}}{2} \\ s \leftarrow s + 0.79 \cdot L_0(V_f) \\ s \end{cases}$$

$$m_0 := M_0(V_f)$$

Уравнения для расчета внутренней энергии воздуха и продуктов сгорания топлива в зависимости от температуры

$$U_{\text{возд}}(t) := 0.0017387 \cdot t^2 + 21.929 \cdot t - 289.14$$

$$U_{\text{нсз}}(t) := 0.0024012 \cdot t^2 + 24.418 \cdot t - 415.43$$

Конец программы расчета характеристик топлива и продуктов сгорания

ПРОГРАММА

теплового расчета дизеля $N=const$ при изменении коэффициента избытка воздуха

Считывание данных матриц

$M_1 := \text{READPRN}("C:\text{MTR_HCDz.prn}")$

$V_f := \text{READPRN}("C:\text{MTR_FUEL.mcd}")$

$\text{Reference}: C:\text{Pp_FUEL.mcd}$

$\text{Reference}: C:\text{Pp_HCD_ALFA.mcd}$

Формат матрицы $Mout \alpha \left(\eta_e \ g_e \ p_e \ T_z \ h_u \ \eta_i \ p_i \ D \ S \ V_h \ \alpha \right) ;$

Результат теплового расчета дизеля $MTR_HCD_ALFA := Mout \alpha$

Запись результатов расчета в структурированный текстовый файл
 $\text{WRITEPRN}("C:\text{MTR_HCD_ALFA.prn}") := MTR_HCD_ALFA$

Конец подпрограммы теплового расчета дизеля

$R_Pp_HCD_ALFA.mcd$

Конец лабораторной работы

(В приведенном примере выполнения лабораторной работы графики, построенные по результатам расчетов не приведены)

Рубежный контроль 1.

Рубежный контроль 2.

Программа курса УМК. Список обязательной и дополнительной литературы.

Обязательная литература:

1. Двигатели внутреннего сгорания. В 3 кн. Кн. 1. Теория рабочих процессов: Учебник для вузов/В.Н. Луканин, К.А. Морозов, А.С. Хачиян и др.; Под ред. В.Н. Луканина. – 2-е изд., перераб и доп. – М.: Высшая школа, 2005. – 479с.
2. Элементы системы автоматизированного проектирования ДВС: Алгоритмы прикладных программ: Учеб. пособие для студентов вузов по специальности «Двигатели внутреннего сгорания»/ Р.М. Петриченко, С.А. Батулин, Ю.Н. Исаков и др.; Под общ. ред. Р.М. Петриченко. – Л.: Машиностроение. Ленинград. отделение, 1990. – 328 с.
3. Гусаков С.В. Тепловой расчет рабочих циклов двигателей внутреннего сгорания: Методическое пособие –М.: Изд-во РУДН, 2004. – 15 с.
4. Кустарев Ю.С., Виноградов Л.В. Система автоматизированного расчета воздухоохладителей автотракторных турбопоршневых двигателей. Учебное пособие. – М.: МГТУ «МАМИ», 2003. – 35 с.
5. Борисов Б.К. Разработка пакетов программ вычислительного типа. – М.:Изд-во Моск. ун-та, 1990. – 124 с.
6. Поршнева С.В., Беленкова И.В. Численные методы на базе *Mathcad*. – СПб.: БХВ-Петербург, 2005. – 464 с.

Дополнительная литература:

1. Виноградов Л.В. Система автоматизированного расчета геометрических характеристик профиля лопатки турбины: Учебно-метод. пособие. – М.: РУДН, 1996. – 35с.
2. Виноградов Л.В. Система автоматизированного расчета геометрических характеристик профиля лопатки турбины и построения решетки профилей: Учебно- метод. пособие. – М.:РУДН, 2002. – 58 с.
3. Виноградов Л.В., Кустарев Ю.С. Расчет термодинамических и геометрических параметров цилиндрической камеры сгорания ГТД. Учебное пособие по курсовому и дипломному проектированию. – М.: МАМИ. – 2001.- 68с.
4. Алексеев Е.Р. Mathcad 12/ Е.Р. Алексеев, О.В. Чеснокова. – М.: НТ Пресс, 2005. – 345 с.
5. Кирьянов Д.В. Самоучитель *Mathcad* .-СПб.: БХВ-Петербург, 2003.- 560 с.
6. Тракторные дизели: Справочник/Б.А. Взоров, А.В. Адамович, А.Г. Арабян и др.: Под общ. ред. Б.А. Взорова. – М.: Машиностроение, 1981. – 535 с.
7. Мартынов А.К. Экспериментальная аэродинамика. Допущено Министерством Высшего Образования СССР в качестве учебника для авиационных высших учебных заведений. – М.: Госиздат оборонной промышленности, 1950. – 475 с.

Программа курса УМК. Темы рефератов

1. Влияние параметров окружающей среды на показатели работы дизеля.
2. Влияние высокогорья на параметры работы дизеля.
3. Особенности работы дизеля в жарком климате.
4. Особенности работы дизеля при низких температурах окружающего воздуха.

Программа курса УМК. Учебный тематический план курса УМК.

Учебная неделя	Виды занятий	
1	Лекция 1 (2 ч)	
2	Лекция 1 (2 ч)	
3	Лекция 2 (2 ч)	
4	Лекция 2 (2 ч)	
5	Лекция 3 (2 ч)	
6	Лекция 3 (2 ч)	
7	Лекция 4 (2 ч)	Лабораторная работа 1. (6 часов)
8	Лекция 4 (2 ч) Рубежный контроль 1. (0,5 часа)	
9	Лекция 5 (2 ч)	Лабораторная работа 2. (6 часов)
10	Лекция 5 (2 ч)	
11	Лекция 6 (2 ч)	(6 часов)
12	Лекция 6 (2 ч) Рубежный контроль 2. (0,5 часа)	
13	Лекция 7 (2 ч)	Лабораторная работа 4. (6 часов)
14	Лекция 7 (2 ч)	
15	Лекция 8 (2 ч)	Лабораторная работа 5. (6 часов)
16	Лекция 8 (2 ч)	
17	Лекция 9 (2 ч)	Лабораторная работа 6. (6 часов)
18	Лекция 9 (2 ч)	
Экзамена- ционная сессия	Экзамен	