



Руководства PRADIS

Laduga

июл. 19, 2026

1	Руководство по установке/удалению ПК Pradis и получению лицензионного ключа	1
1.1	Поддерживаемые операционные системы	1
1.2	Установка и удаление PRADIS для Windows	1
1.3	Установка и удаление PRADIS для Astra Linux	17
2	Практическое введение в PRADIS	23
2.1	Введение	23
2.2	Что такое PRADIS	24
2.3	Подготовка к работе	26
2.4	Пример: простая механическая схема	29
2.5	Алгоритм по созданию расчетной схемы	40
2.6	Контрольный список первого самостоятельного расчёта	43
2.7	Типичные ошибки	44
2.8	Заключение	45
3	Руководство пользователя PRADIS	49
3.1	Руководство по работе с препроцессором	49
3.2	Руководство по управлению вычислительным ядром Pradis	53
3.3	Степени свободы, сплиттеры и методы построения многодоменных схем	62
3.4	Руководство по включению в библиотеки комплекса PRADIS моделей элементов и объектов	81

Руководство по установке/удалению ПК Pradis и получению лицензионного ключа

В данном разделе приведены инструкции по установке, удалению и обновлению программного комплекса PRADIS, включая процесс запроса и активации лицензионного ключа на поддерживаемых операционных системах.

1.1 Поддерживаемые операционные системы

На текущий момент установка PRADIS официально поддерживается для следующих операционных систем:

Таблица 1: Поддерживаемые операционные системы

Операционная система	Версия / Архитектура
Windows	7 и выше (x86 / x64)
Astra Linux	(amd64)

Предупреждение

Если требуется использовать PRADIS на другой платформе, пожалуйста, обратитесь к официальному дистрибьютору для получения информации о возможных сроках расширения списка поддерживаемых ОС.

1.2 Установка и удаление PRADIS для Windows

1.2.1 Минимальные системные требования

Параметр	Значение
Операционная система	Windows 7 и выше
Архитектура	x86 / x64
Объем оперативной памяти	4 ГБ (рекомендуется 8 ГБ)
Свободное место на диске	750 МБ (основной модуль)

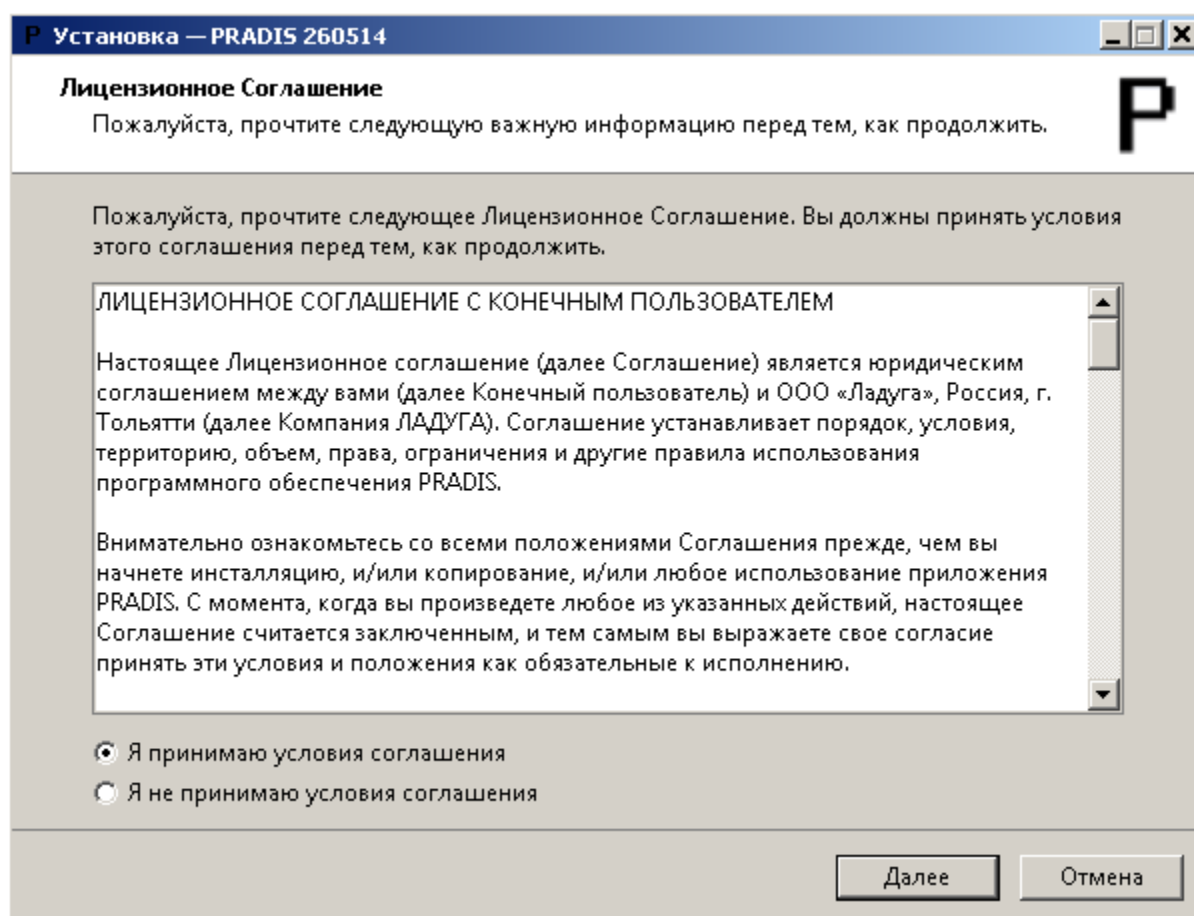
1.2.2 Установка Pradis

Шаг 1. Запуск установщика

Запустите файл установщика **PRADISSetup.ru.X.X.XXXXXX.exe** от имени администратора (правый клик → «Запуск от имени администратора»).

Шаг 2. Принятие лицензионного соглашения

Ознакомьтесь с условиями лицензионного соглашения. Для продолжения установки необходимо принять его условия.

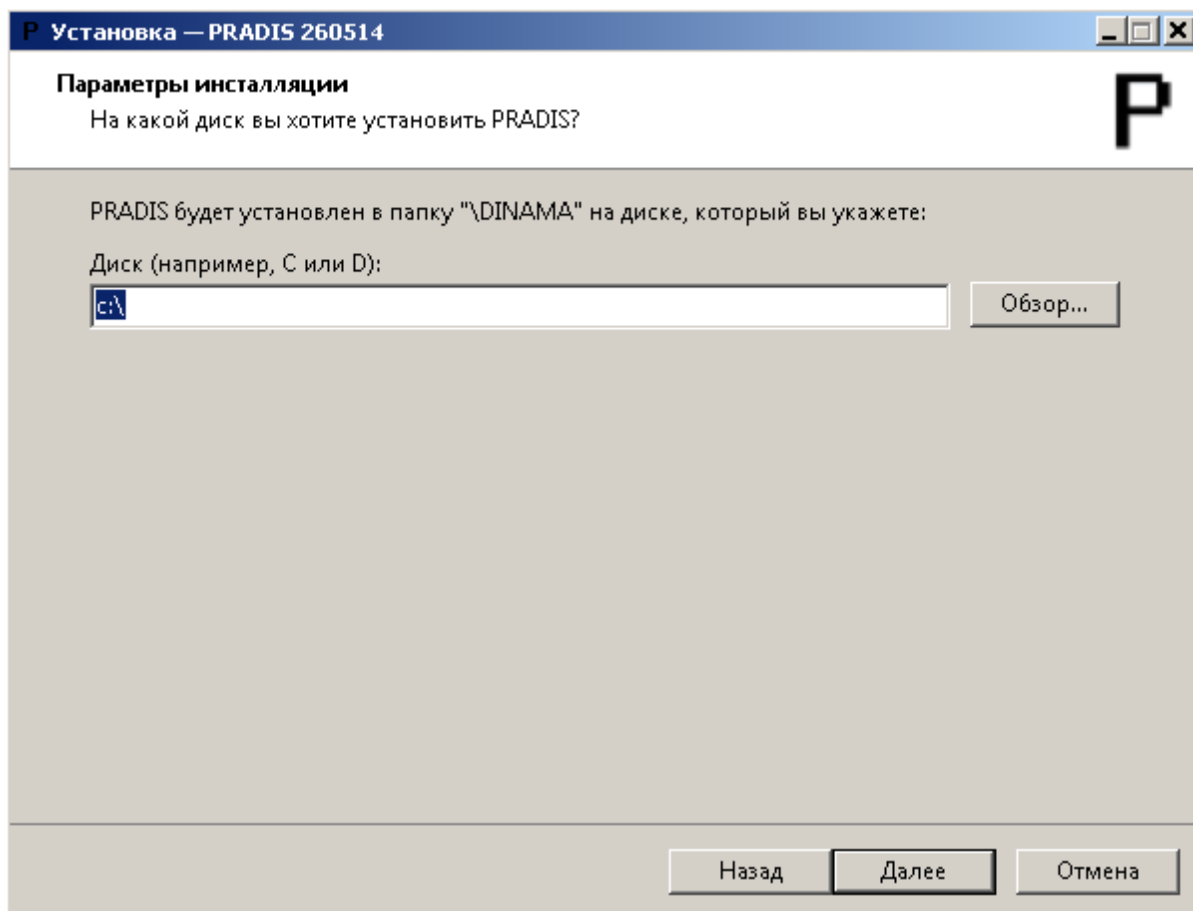


Шаг 3. Выбор каталога установки

По умолчанию программа устанавливается в каталог:

C:\DINAMA

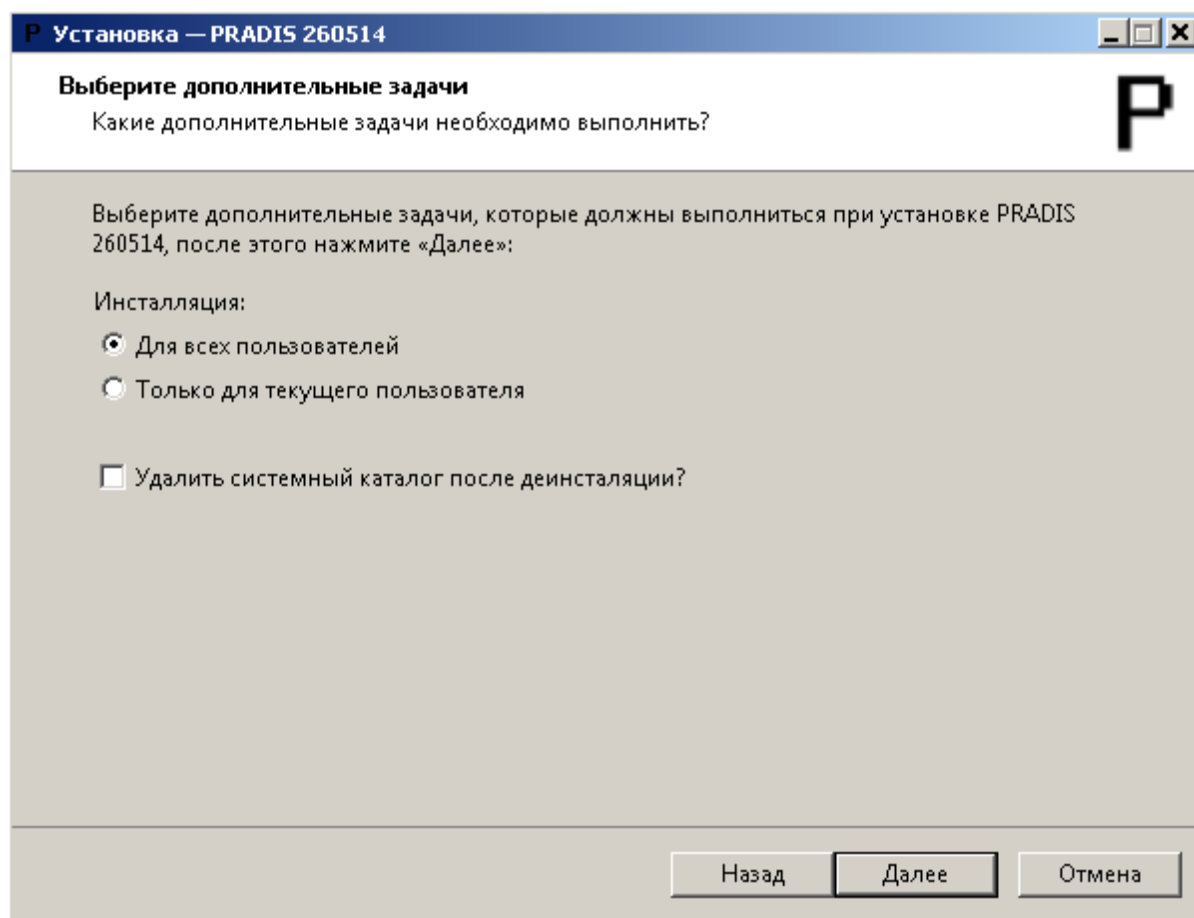
Для этого, оставить путь, заданный по умолчанию, без изменений, как показано на рисунке



При необходимости можно изменить путь установки.

Шаг 4. Выбор дополнительных параметров установки

После выбора каталога установщик предложит настроить дополнительные параметры .



В данном окне доступны следующие опции:

1. Выбор типа инсталляции

Таблица 2: Выбор типа инсталляции

Опция	Рекомендация
Для всех пользователей	Рекомендуется. Программа устанавливается в C:\Program Files\... и становится доступна всем учётным записям компьютера. Требуется права администратора.
Только для текущего пользователя	Выбирать только в случае отсутствия прав администратора или для тестовой установки. Программа будет установлена в папку текущего пользователя и не будет видна другим.

2. Удаление системного каталога после деинсталляции

По умолчанию данная опция **выключена** (галочка снята).

Таблица 3: Когда следует изменять настройку опции

Состояние опции	Когда следует выбирать
Оставьте опцию выключенной (снимите галочку)	• Вы планируете в будущем обновить или переустановить PRADIS • В системном каталоге хранятся важные библиотеки, настройки или пользовательские проекты • Вы хотите иметь возможность отката
Включите опцию (установите галочку)	• Вы полностью удаляете PRADIS с компьютера и не планируете использовать его снова • Требуется выполнить «чистую» переустановку для устранения проблем • Место на диске критично, а системный каталог занимает много места

Примечание**Что такое «системный каталог PRADIS»?**

Это папка, в которой программа хранит служебные данные: конфигурационные файлы, библиотеки моделей, настройки лицензий, а иногда и пользовательские проекты. Её удаление при деинсталляции может привести к потере данных, поэтому данная опция должна использоваться осознанно.

После выбора необходимых опций нажмите кнопку «**Далее**» для продолжения установки.

Шаг 5. Сохранение системного каталога предыдущей версии (при обновлении)

Если на компьютере обнаружена предыдущая версия PRADIS, установщик предложит выбрать, сохранять ли системный каталог.

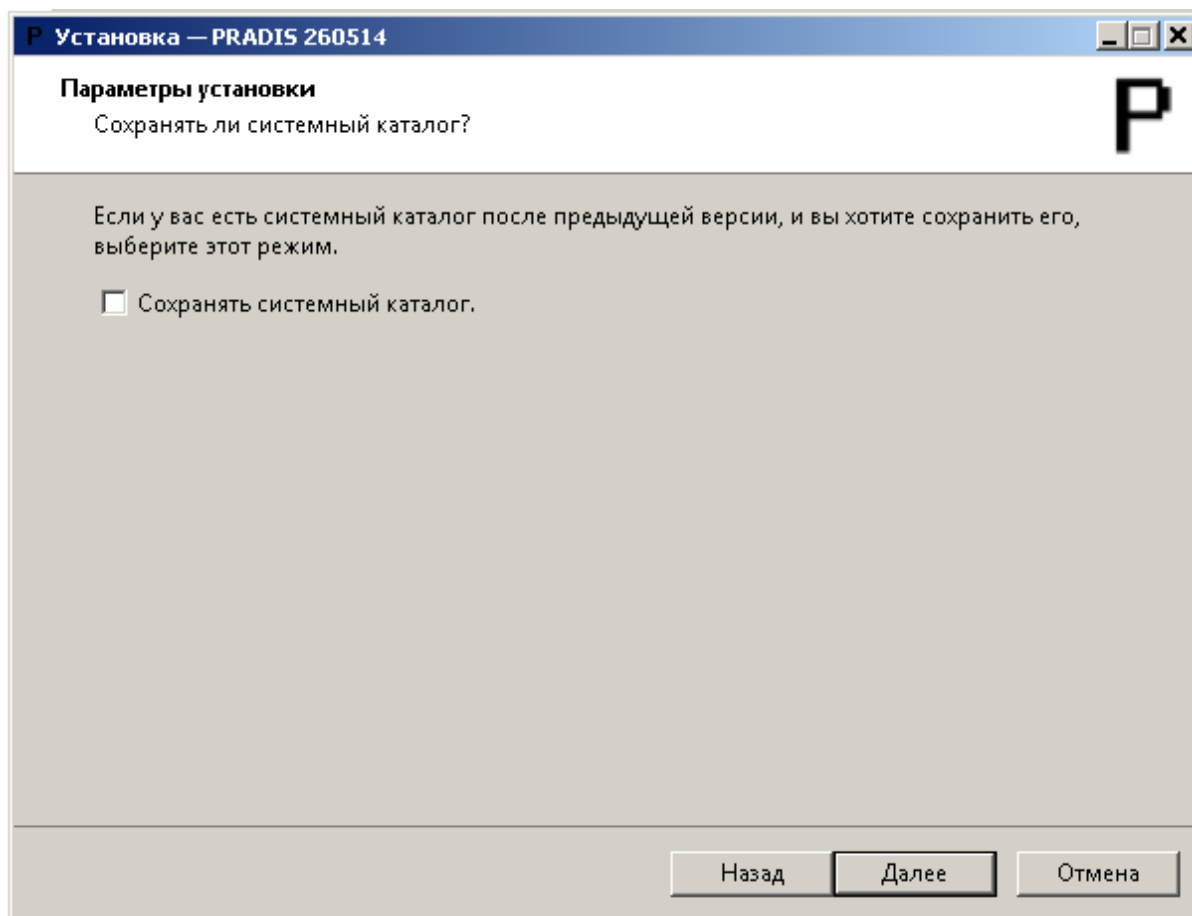


Таблица 4: Когда следует сохранять системный каталог

Состояние опции	Когда следует выбирать
☐ Сохранять системный каталог (установить галочку)	• Вы обновляете PRADIS с предыдущей версии • В системном каталоге хранятся важные настройки, библиотеки или пользовательские данные • Вы хотите избежать потери настроек и не тратить время на перенастройку программы
☐ Сохранять системный каталог (снять галочку)	• Вы выполняете «чистую» установку без сохранения старых данных • Предыдущая версия работает некорректно или повреждена • Вы хотите полностью удалить старые файлы и освободить место на диске

Примечание**Что произойдёт при сохранении системного каталога?**

- Все настройки, библиотеки моделей и пользовательские данные из старой версии будут перенесены в новую
- Вам не придётся заново настраивать программу
- Пользовательские проекты останутся доступными

Что произойдёт, если снять галочку?

- Системный каталог предыдущей версии будет удалён или перезаписан
- Все старые настройки и данные будут потеряны
- После установки потребуется выполнить первоначальную настройку программы заново

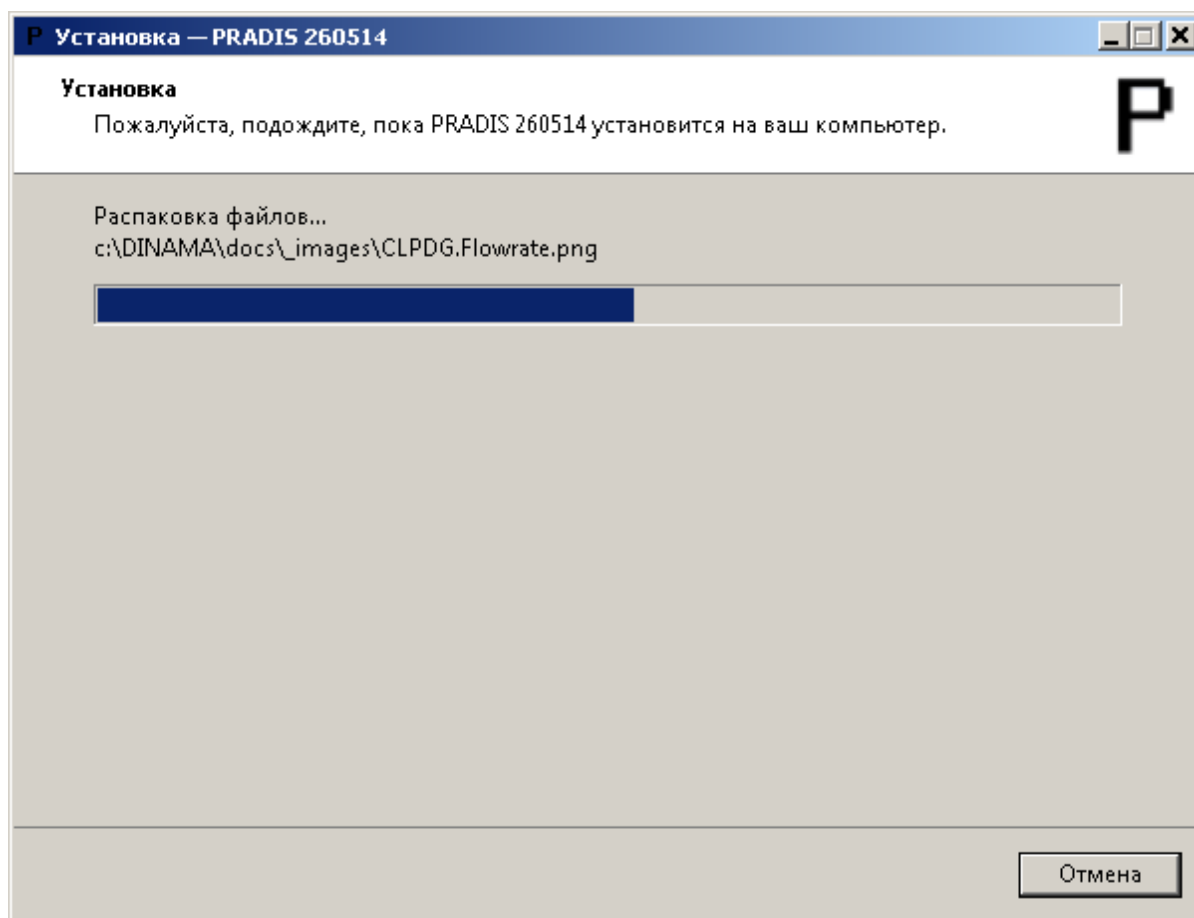
Предупреждение

Если предыдущая версия PRADIS работает с ошибками или повреждена, **рекомендуется снять галочку** — сохранение проблемного каталога может перенести ошибки в новую версию.

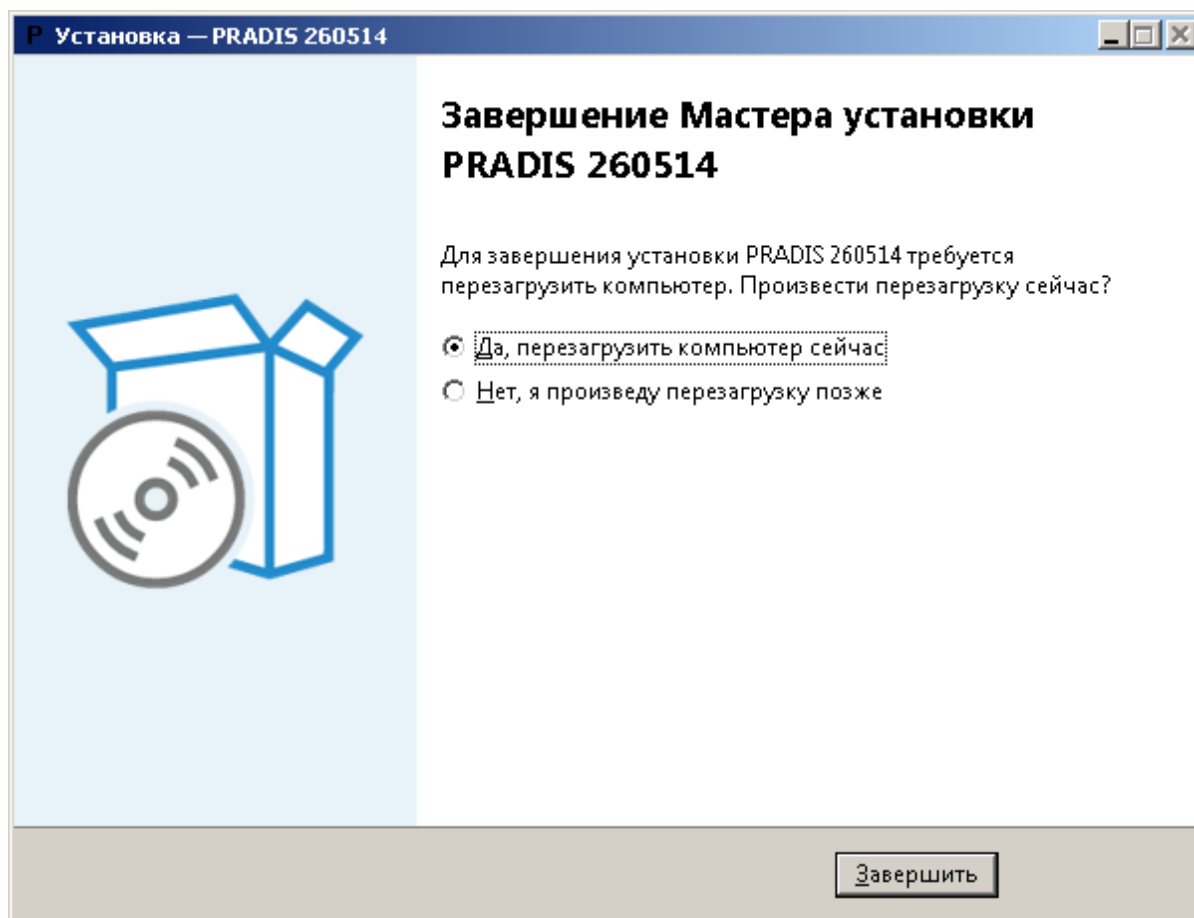
После выбора необходимой опции нажмите кнопку «Далее» для продолжения установки.

Шаг 6. Завершение установки

Дождитесь окончания копирования файлов.



При необходимости перезагрузите компьютер (установщик предложит это сделать автоматически).



1.2.3 Получение и установка лицензионного файла

Pradis работает на основе локальной однопользовательской программной лицензии.

Запрос на получение лицензионного файла

1. Откройте каталог, в который установлена программа PRADIS (в данном примере - C:DINAMA)

Компьютер ▾ Локальный диск (C:) ▾ DINAMA ▾				
Поиск: DINAMA				
Открыть ▾ Записать на оптический диск Новая папка				
Имя ^	Дата изменения	Тип	Размер	
app	29.05.2026 17:15	Папка с файлами		
data	29.05.2026 17:15	Папка с файлами		
dll	29.05.2026 17:29	Папка с файлами		
docs	29.05.2026 17:29	Папка с файлами		
examples	29.05.2026 17:17	Папка с файлами		
icons	29.05.2026 17:34	Папка с файлами		
license	29.05.2026 17:31	Папка с файлами		
logs	29.05.2026 17:31	Папка с файлами		
plugin	29.05.2026 17:31	Папка с файлами		
pme	29.05.2026 17:31	Папка с файлами		
post	29.05.2026 17:31	Папка с файлами		
pradis32	29.05.2026 17:31	Папка с файлами		
prepradis	29.05.2026 17:17	Папка с файлами		
Python	29.05.2026 17:32	Папка с файлами		
Qucs	29.05.2026 17:17	Папка с файлами		
sysarm	29.05.2026 17:33	Папка с файлами		
temp	29.05.2026 17:34	Папка с файлами		
templates	29.05.2026 17:19	Папка с файлами		
get_params	14.05.2026 13:50	Приложение	61 КБ	
get_params	29.05.2026 21:30	Текстовый документ	1 КБ	
gui	14.05.2026 13:50	Приложение	10 021 КБ	
parameters	29.05.2026 21:30	Исходный файл JSON	1 КБ	
unins000	29.05.2026 17:34	Postprocessor Dat File	6 122 КБ	
P unins000	29.05.2026 17:28	Приложение	4 264 КБ	

2. Запустите файл `get_params.exe`
3. Дождитесь появления файла `parameters.json` в том же каталоге
4. Сгенерированный файл `parameters.json` необходимо передать по электронной почте в техподдержку PRADIS или лично менеджеру, предоставившему дистрибутив PRADIS.

Контактные данные по вопросам предоставления лицензионного ключа: Кубышкин Вадим

- **Кому:** kvi@laduga.com
- **Тема письма:** Запрос лицензии PRADIS / [ваше имя или организация]

Компьютер ▾ Локальный диск (C:) ▾ DINAMA ▾				Поиск: DINAMA
Открыть ▾ Записать на оптический диск Новая папка				
Имя ^	Дата изменения	Тип	Размер	
app	29.05.2026 17:15	Папка с файлами		
data	29.05.2026 17:15	Папка с файлами		
dll	29.05.2026 17:29	Папка с файлами		
docs	29.05.2026 17:29	Папка с файлами		
examples	29.05.2026 17:17	Папка с файлами		
icons	29.05.2026 17:34	Папка с файлами		
license	29.05.2026 17:31	Папка с файлами		
logs	29.05.2026 17:31	Папка с файлами		
plugin	29.05.2026 17:31	Папка с файлами		
pme	29.05.2026 17:31	Папка с файлами		
post	29.05.2026 17:31	Папка с файлами		
pradis32	29.05.2026 17:31	Папка с файлами		
prepradis	29.05.2026 17:17	Папка с файлами		
Python	29.05.2026 17:32	Папка с файлами		
Qucs	29.05.2026 17:33	Папка с файлами		
sysarm	29.05.2026 17:33	Папка с файлами		
temp	29.05.2026 17:34	Папка с файлами		
templates	29.05.2026 17:19	Папка с файлами		
get_params	14.05.2026 13:50	Приложение	61 КБ	
get_params	29.05.2026 21:30	Текстовый документ	1 КБ	
gui	14.05.2026 13:50	Приложение	10 021 КБ	
parameters	29.05.2026 21:30	Исходный файл JSON	1 КБ	
unins000	29.05.2026 17:34	Postprocessor Dat File	6 122 КБ	
P unins000	29.05.2026 17:28	Приложение	4 264 КБ	

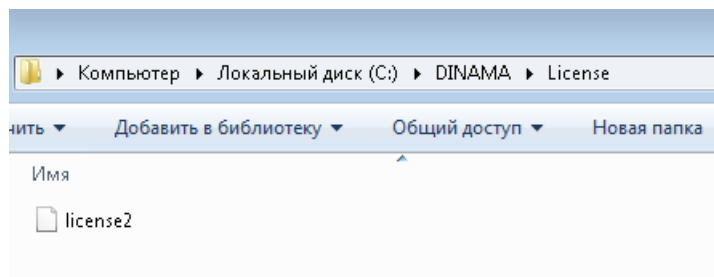
Примечание

Не изменяйте содержимое файла `parameters.json` вручную. Это может привести к некорректной генерации лицензионного ключа.

5. После обработки запроса вы получите лицензионный файл (`license2`).

Установка лицензионного ключа

1. Сохраните вложенный файл `license2` из письма на жёсткий диск
2. Полученный Файл `license2` необходимо поместить в папку `DINAMA\license`



Проверка установки лицензионного файла

После успешной активации лицензии:

- В меню **Справка** → **О программе** (или **Help** → **About**) отображается информация о действующей лицензии
- Все функции PRADIS становятся доступны для использования

Также информацию о лицензии можно увидеть во время выполнения расчета в консоли

```
+-----+
| LADUGA LLC                               |
| www.laduga.com                           |
| www.pradis.laduga.com                     |
+-----+
| PRADIS, A Program for Analysis of Dinamic Systems |
| Version: X.X.XXXXXX                         |
| Licensed to: XXXXXX: XXXXXXXXXXXXXXXXXXXX    |
| License type: Professional license           |
| License expire date: YYYY-MM-DD             |
| You have XXX license days left               |
+-----+
```

Примечание

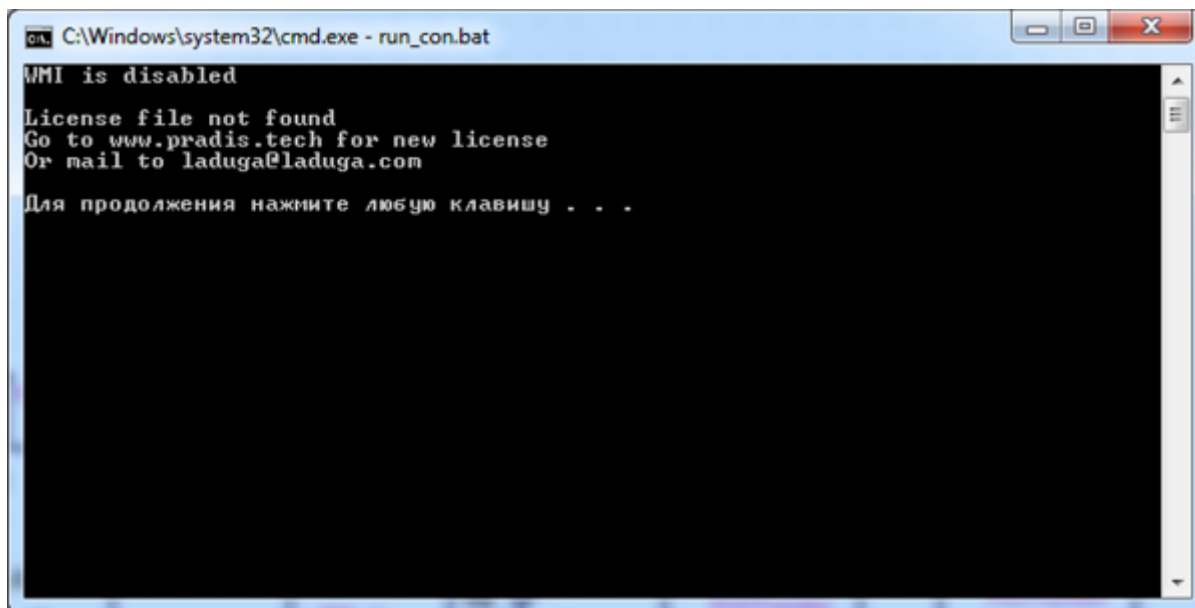
Лицензионный файл license2 привязан к конкретному компьютеру (аппаратной конфигурации). При замене оборудования или переустановке операционной системы может потребоваться повторное получение лицензии.

Предупреждение

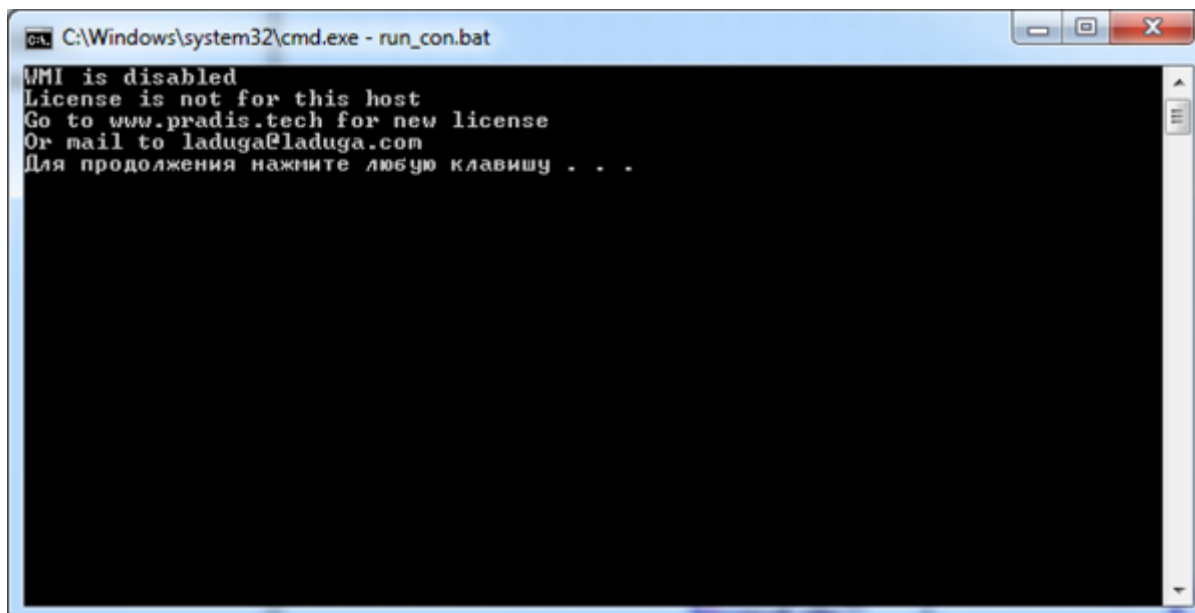
- Не удаляйте и не перемещайте файл license2 после активации
- Не изменяйте содержимое лицензионного файла вручную
- При утере лицензионного файла обратитесь в службу поддержки для его восстановления

Возможные ошибки:

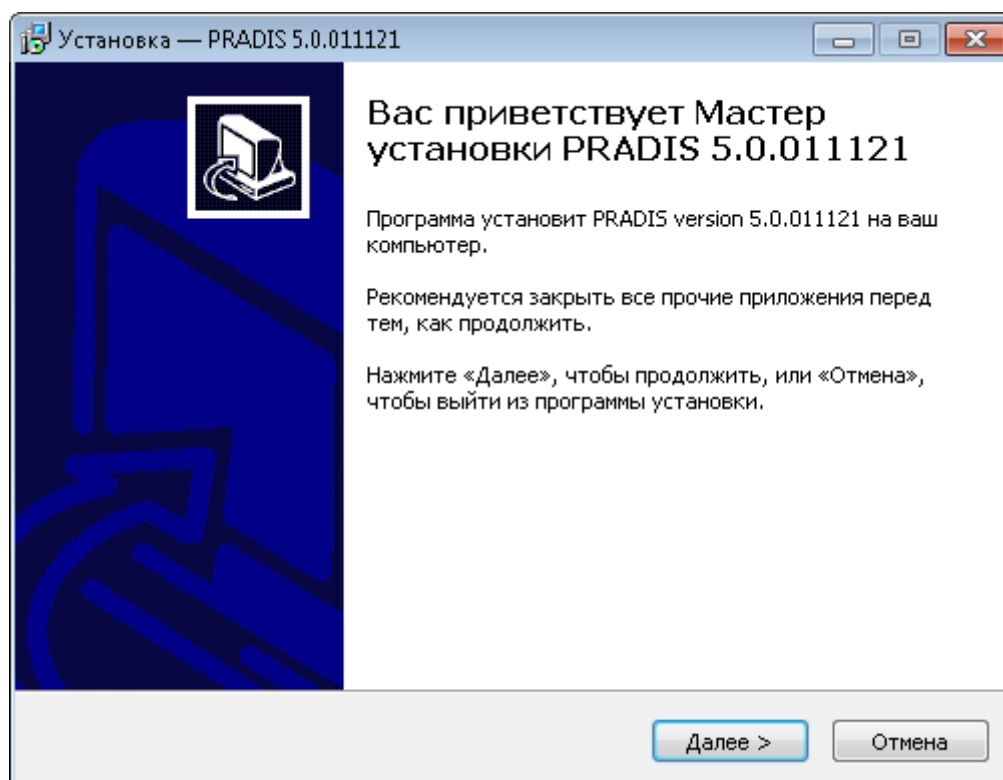
License file not found - файл с лицензией отсутствует или добавлен в некорректную папку.



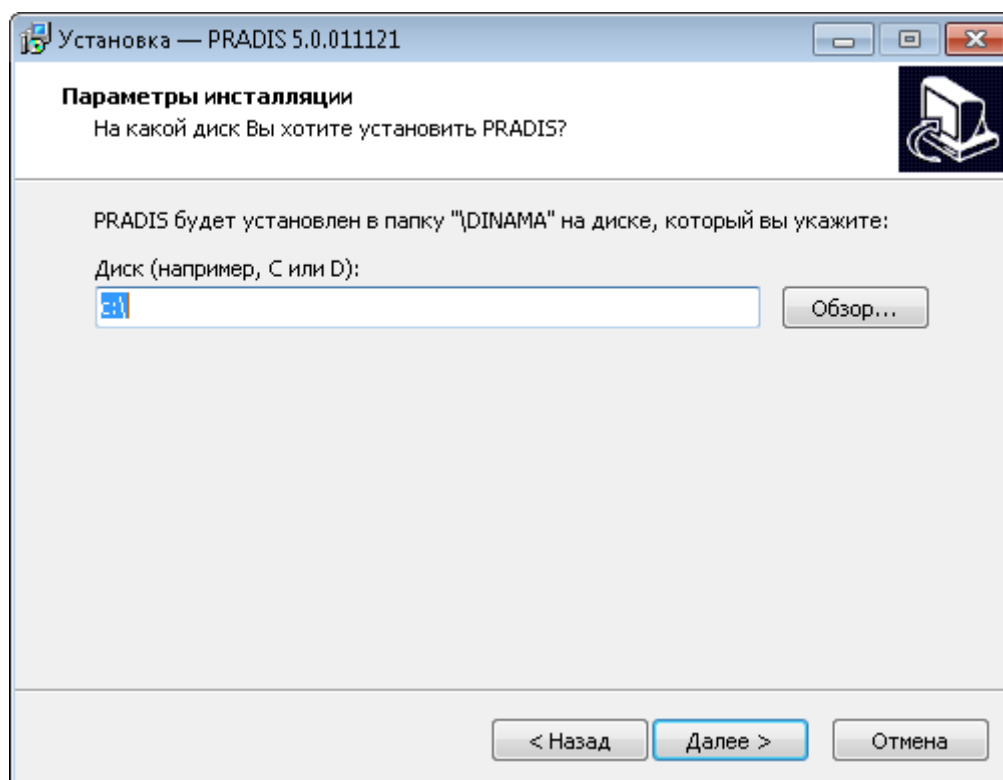
License not for this host - файл лицензии некорректен, обратитесь в техподдержку PRADIS.

**1.2.4 Обновление ПК PRADIS**

1. Запустите файл установки обновления PRADISUpdate.X.X.XXXXXX.exe.
2. В приветственном окне нажмите «Далее».

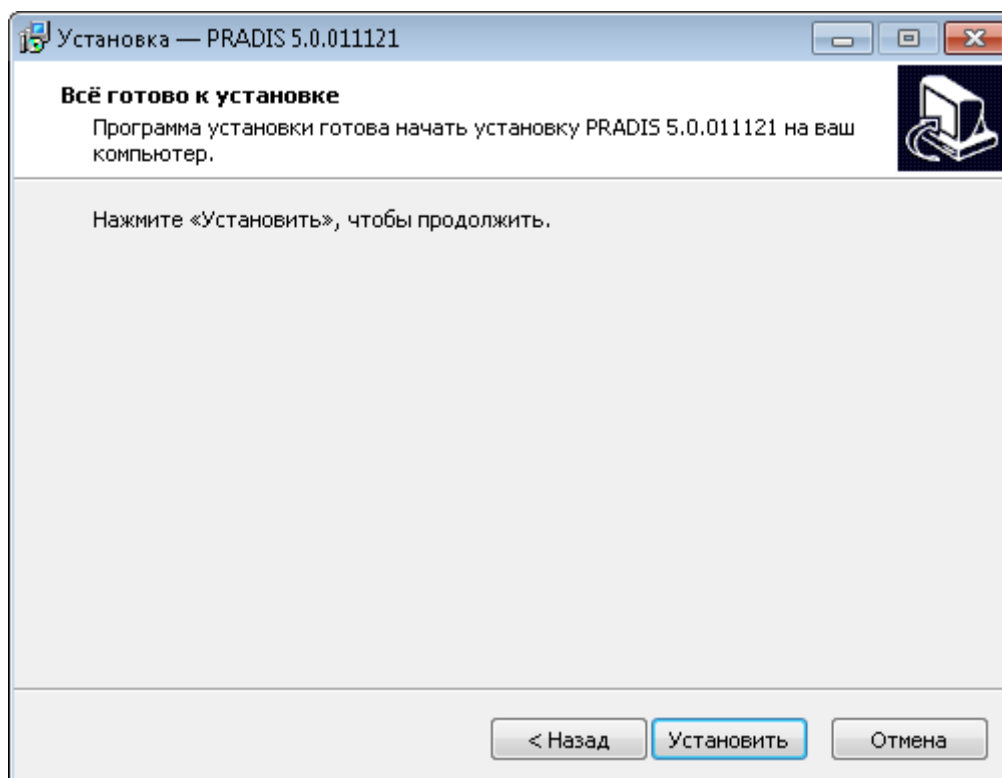


3. Выберите диск, на котором находится папка DINAMA в которой установлен PRADIS.

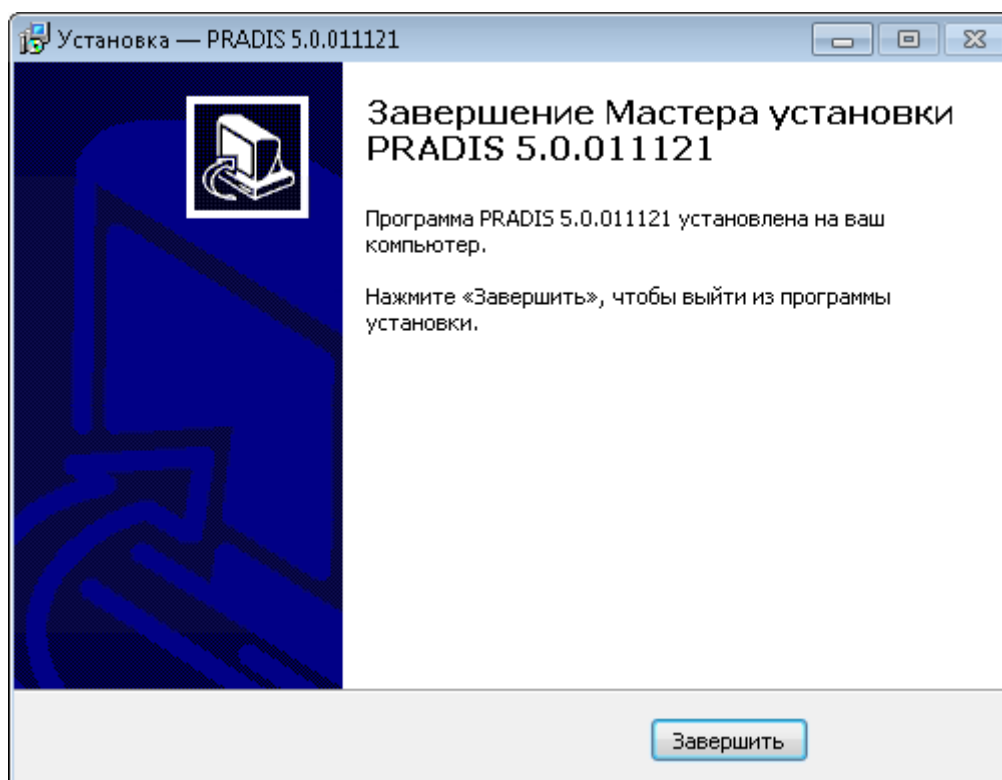


4. Нажмите «Далее».

5. Нажмите «Установить».



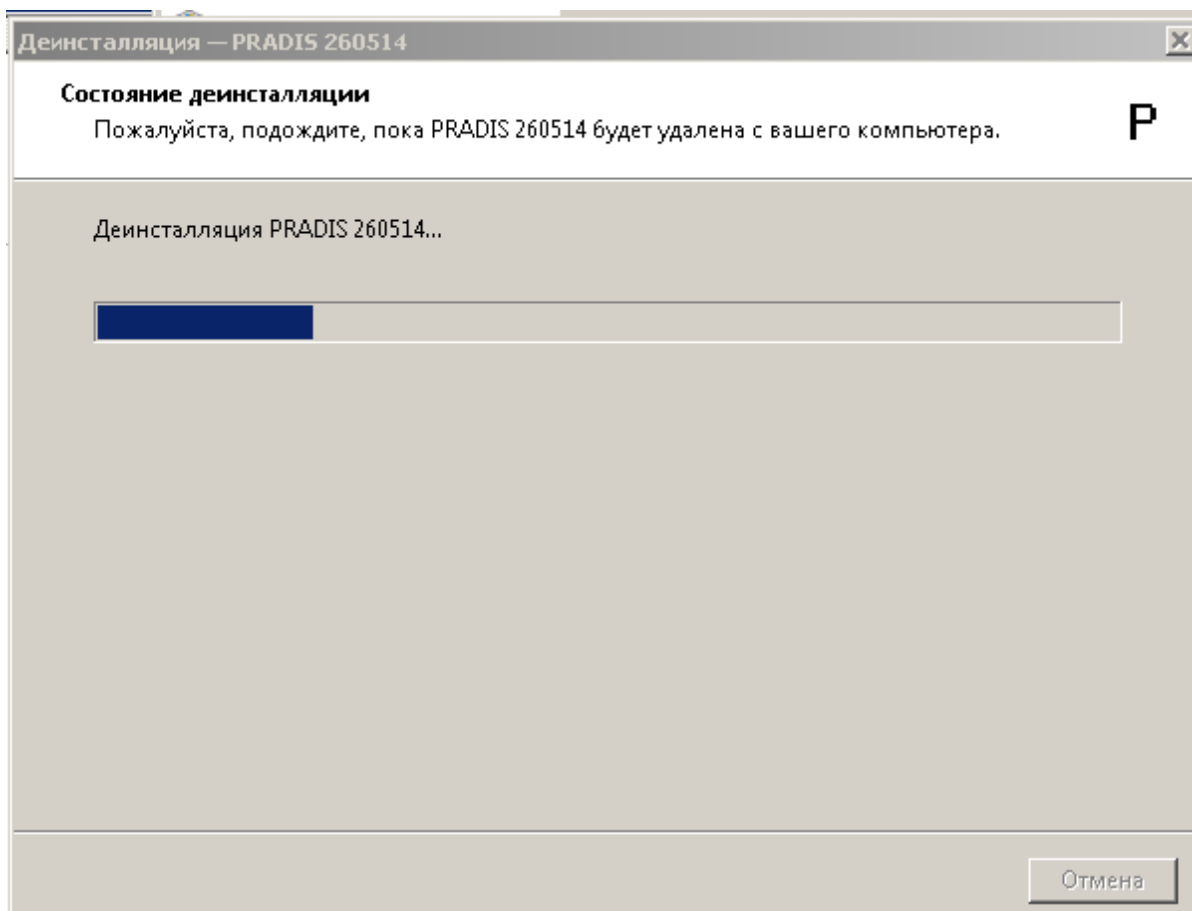
6. Для завершения установки нажмите «Завершить».



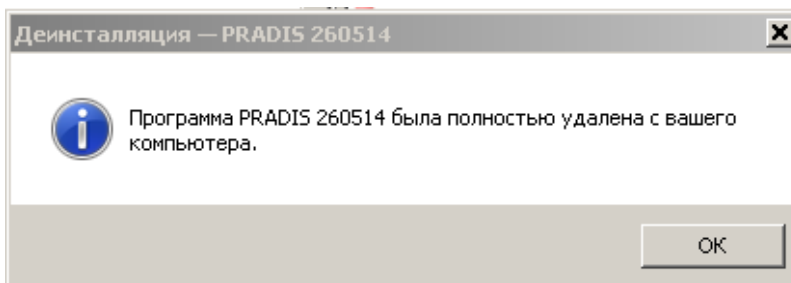
1.2.5 Удаление ПК Pradis

Способ 1

1. Откройте **Панель управления** → **Программы и компоненты** (или «Установка и удаление программ»)
2. Найдите в списке **PRADIS** (или «Pradis — Программа анализа динамических систем»)
3. Выделите программу и нажмите кнопку «Удалить». откроется окно удаления ПО Pradis



4. Следуйте инструкциям мастера деинсталляции. Окно с результатом удаления ПО Pradis



Способ 2

1. Зайдите в папку, в которую установлен PRADIS, например, C:DINAMA, и запустите unins000.exe.
2. Подтвердите желание удалить PRADIS.

1.3 Установка и удаление PRADIS для Astra Linux**1.3.1 Минимальные системные требования**

Параметр	Значение
Операционная система	Astra Linux (amd64)
Архитектура	amd64
Объем оперативной памяти	4 ГБ (рекомендуется 8 ГБ)
Свободное место на диске	750 МБ (основной модуль)

1.3.2 Подготовка к установке

Перед установкой проверьте, что выполнены следующие условия:

- используется **Astra Linux** на архитектуре amd64
- у пользователя есть права администратора (доступ к команде sudo)
- файл пакета PRADIS имеет вид pradis_<версия>_amd64-astralinux.deb
- в системе настроены репозитории Astra Linux или доступен другой источник пакетов для установки зависимостей

Зависимости

Для работы PRADIS требуются следующие системные библиотеки и компоненты:

Категория	Компоненты
Системные библиотеки	libc6, libgcc-s1, libstdc++6
Инструменты и интерпретаторы	gcc, python3
Библиотеки Qt5	libqt5core5a, libqt5gui5, libqt5widgets5
Дополнительные модули Qt5	libqt5printsupport5, libqt5svg5

Примечание

При установке через apt зависимости устанавливаются автоматически, если они доступны в подключенных репозиториях.

Проверка установки необходимых библиотек

Выполните в командной строке:

```
dpkg -l libc6 libgcc-s1 libstdc++6 gcc python3 libqt5core5a libqt5gui5 \
↳ libqt5widgets5 libqt5printsupport5 libqt5svg5
```

Интерпретация результата

Код в начале строки	Значение
ii	Библиотека установлена □
un или rp	Пакет не установлен или не найден □
rc	Пакет удалён, но остались конфигурационные файлы

Пример вывода (фрагмент)

```
ii  libc6                2.31-13+deb11u5 amd64      GNU C Library
ii  libgcc-s1            10.2.1-6         amd64      GCC support library
un  libqt5svg5           <none>           <none>     (no description
↳ available)
```

В данном примере libqt5svg5 отсутствует — его необходимо установить перед продолжением.

Дополнительная проверка (опционально)

Для более детальной проверки можно использовать:

```
# Проверка наличия .so файлов в системе
ldconfig -p | grep -E "libc\.so|libgcc_s\.so|libstdc++\.so|libQt5"

# Проверка версий Python и GCC
python3 --version
gcc --version
```

Установка отсутствующих зависимостей

Если каких-либо пакетов не хватает, установите их:

```
sudo apt update
sudo apt install libc6 libgcc-s1 libstdc++6 gcc python3 \
  libqt5core5a libqt5gui5 libqt5widgets5 \
  libqt5printsupport5 libqt5svg5
```

После установки рекомендуется повторить проверку командой `dpkg -l ....`

1.3.3 Установка PRADIS

Шаг 1. Открытие терминала

Откройте терминал (например, через меню приложений или сочетанием клавиш Ctrl+Alt+T).

Шаг 2. Переход в каталог с установочным пакетом

Перейдите в каталог, где находится полученный .deb-пакет PRADIS. Например, если пакет сохранён в папке Загрузки:

```
cd ~/Загрузки
```

или

```
cd ~/Downloads
```

Шаг 3. Установка пакета

Выполните команду установки:

```
sudo apt install ./pradis_<версия>_amd64-astralinux.deb
```

Примечание

Вместо pradis_<версия>_amd64-astralinux.deb укажите фактическое имя полученного файла. Например: pradis_1.2.56789_amd64-astralinux.deb

Шаг 4. Подтверждение установки

Если система запросит подтверждение, введите Y (или Д) и нажмите клавишу Enter.

Шаг 5. Завершение установки

Дождитесь окончания установки. После завершения установки **перелогиньтесь в систему** или откройте новый терминал. Это нужно, чтобы применились переменные окружения PRADIS.

Альтернативный способ установки (через dpkg)

Если установка выполняется на машине без доступа к репозиториям, можно использовать команду dpkg:

```
sudo dpkg -i pradis_<версия>_amd64-astralinux.deb
```

Если после этого появилось сообщение о ненастроенных зависимостях, выполните:

```
sudo apt install -f
```

После успешного выполнения команды пакет PRADIS будет настроен системой.

Что устанавливается?

Пакет устанавливает PRADIS в каталог /opt/pradis и добавляет системные команды запуска:

- pradis-pq — препроцессор PRADIS
- pradis-post — постпроцессор PRADIS

- pradis-python — запуск Python с окружением PRADIS
- pradis-sch-to-pa8 — конвертация схем .sch для PA8

Также в графическое меню добавляются ярлыки **PRADIS Pre** и **PRADIS Post**.

Основной каталог установленной системы:

```
/opt/pradis/DINAMA
```

Примеры PRADIS расположены в каталоге:

```
/opt/pradis/DINAMA/examples
```

1.3.4 Проверка установки

Проверьте, что пакет установлен:

```
dpkg -s pradis
```

Проверьте наличие команд запуска:

```
which pradis-pq
which pradis-post
which pradis-python
which pradis-sch-to-pa8
```

Запустить препроцессор и постпроцессор можно из графического меню Astra Linux или из терминала:

```
pradis-pq
pradis-post
```

Примечание

Дополнительная настройка лицензии для Astra Linux-пакета **не требуется**. Лицензионная информация обычно уже встроена в пакет или настраивается отдельно дистрибьютором.

1.3.5 Обновление PRADIS

Если дистрибьютор предоставил новый .deb-пакет, установите его той же командой:

```
sudo apt install ./pradis_<новая_версия>_amd64-astralinux.deb
```

Система автоматически заменит установленную версию PRADIS на новую.

1.3.6 Удаление PRADIS

Способ 1 (через apt)

Для удаления PRADIS выполните:

```
sudo apt remove pradis
```

Команда удаляет установленный пакет и системные ярлыки. Пользовательские файлы, созданные вне каталога установки, не удаляются.

Способ 2 (полное удаление с конфигурацией)

Если требуется удалить пакет вместе с системными конфигурационными файлами, выполните:

```
sudo apt purge pradis
```

Дополнительная очистка

После удаления можно также удалить кэш журналов текущего пользователя:

```
rm -rf ~/.cache/pradis
```

Если каталог `/opt/pradis` остался после удаления, проверьте, что в нём нет нужных пользовательских файлов, и удалите его вручную:

```
sudo rm -rf /opt/pradis
```

Практическое введение в PRADIS

2.1 Введение

2.1.1 Целевая аудитория

Данный документ предназначен для инженеров, исследователей, преподавателей и студентов, которые впервые знакомятся с программным комплексом **PRADIS** и желают в кратчайшие сроки освоить основные принципы построения моделей и выполнения расчётов.

Руководство ориентировано на пользователей, не имеющих опыта работы с PRADIS. В нём рассматривается базовый цикл моделирования: создание проекта, построение расчётной схемы, выбор компонентов, настройка параметров, запуск расчёта и анализ полученных результатов.

Для работы с данным руководством достаточно общих знаний в области физики и инженерного моделирования. Углублённое изучение математических методов, многомерного моделирования и особенностей внутренней архитектуры PRADIS в документ не входит и рассматривается в специализированной документации.

2.1.2 Ожидаемые результаты и приобретаемые навыки

После изучения данного руководства пользователь сможет:

- создавать новые проекты и расчётные схемы;
- выбирать необходимые библиотеки компонентов в зависимости от физического домена модели;
- собирать простые расчётные схемы из библиотечных элементов;
- соединять компоненты через совместимые порты;
- задавать параметры элементов, начальные и граничные условия;
- настраивать параметры моделирования и запускать расчёт;

- анализировать результаты моделирования с помощью постпроцессора;
- строить графики, таблицы и другие средства визуализации результатов;
- выполнять простую проверку корректности модели и интерпретировать полученные результаты.

В результате пользователь освоит базовый цикл работы в PRADIS — от создания проекта до анализа результатов моделирования — и сможет самостоятельно создавать и исследовать несложные инженерные модели.

2.1.3 Какие разделы полного руководства читать дальше

Настоящее руководство знакомит с базовыми возможностями PRADIS и позволяет освоить основные принципы работы с программным комплексом. Для более глубокого изучения рекомендуется обратиться к следующим материалам:

1. **Полное руководство пользователя PRADIS — подробное описание** интерфейса, режимов работы и возможностей программы.
2. **Справочник по библиотекам элементов — описание библиотек** компонентов и их назначения.
3. **Описание компонентов библиотек — подробные сведения о назначении,** портах, параметрах и физических моделях отдельных элементов.
4. **Руководство по работе с подсхемами и объектами — создание** повторно используемых моделей, организация сложных проектов и работа с пользовательскими объектами.
5. **Руководство по математическим методам PRADIS — описание** математической модели, методов численного интегрирования и принципов формирования системы уравнений.
6. **Руководство по разработке пользовательских компонентов — создание** собственных моделей элементов и их интеграция в программный комплекс.
7. **Руководство по подключению пользовательских библиотек и внешних программ** — расширение функциональности PRADIS и взаимодействие с внешними приложениями.

После освоения материалов данного руководства рекомендуется в первую очередь ознакомиться со **Справочником по библиотекам элементов** и **Описанием компонентов библиотек**, так как именно они чаще всего используются при построении собственных расчётных схем.

2.2 Что такое PRADIS

2.2.1 Назначение комплекса

PRADIS — программный комплекс для математического моделирования и анализа динамического поведения технических систем различной физической природы. Он позволяет создавать расчётные модели, выполнять их численное решение и анализировать полученные результаты.

В состав комплекса входят препроцессор, вычислительное ядро (решатель) и постпроцессор, обеспечивающие полный цикл моделирования — от построения модели до анализа результатов.

Препроцессор PRADIS Qucs представляет собой интерактивную графическую среду, предназначенную для создания, редактирования и параметризации расчётных схем. Пользователь собирает модель из готовых библиотечных компонентов, задаёт параметры элементов, определяет начальные и граничные условия, после чего подготавливает модель к расчёту.

На основе построенной расчётной схемы PRADIS автоматически формирует математическую модель в машиночитаемом виде и передаёт её вычислительному ядру для выполнения численного моделирования. Результаты расчёта затем анализируются с помощью постпроцессора.

2.2.2 Минимальный словарь

Понятие	Практический смысл
Проект	Рабочая единица пользователя, в которой хранятся схемы, подсхемы, объекты, исходные данные, настройки и результаты расчётов.
Схема	Графическое представление моделируемой системы, состоящее из элементов, портов, связей и условий расчёта.
Элемент	Базовый компонент расчётной схемы. Имеет порты, параметры и внутреннюю математическую модель, описывающую его физическое поведение.
Библиотека элементов	Набор компонентов, сгруппированных по физическому домену или назначению (механика, гидравлика, пневматика, электротехника, управление и др.).
Порт	Интерфейс элемента, через который он может быть подключён к другим компонентам модели. Определяет тип передаваемых физических величин или сигналов.
Связь	Логическое соединение между портами элементов, которое задаёт их взаимодействие в модели. Связь использует порты, но не является тем же самым, что порт.
Узел	Общая точка соединения одного или нескольких портов, имеющих одинаковый тип и одинаковое количество степеней свободы.
Степень свободы (DOF)	Независимая координата или физическая переменная, описывающая состояние системы, например, перемещение, угол поворота, давление или температура.
Параметр	Численное или структурное свойство элемента: масса, длина, жёсткость, коэффициент демпфирования, давление, координаты, свойства материала и другие характеристики.
Переменная	Физическая величина, вычисляемая в процессе моделирования. Например, скорость, ускорение, сила, расход, давление или температура.
Подсхема	Логически завершённый фрагмент модели, который можно использовать как самостоятельный компонент в других схемах.
Объект	Пользовательская структура, которая может содержать одну или несколько моделей, подсхем и собственную внутреннюю логику.
Математическая модель	Система уравнений, автоматически сформированная PRADIS на основе расчётной схемы и параметров компонентов.
Препроцессор	Часть PRADIS, которая анализирует расчётную схему, проверяет её корректность, формирует внутреннюю структуру модели и подготавливает её к расчёту.
Постпроцессор	Средства просмотра, анализа и визуализации результатов моделирования: графики, таблицы, анимация, экспорт данных и другие инструменты анализа.

2.2.3 Расчётная схема, элементы, порты и связи

Расчётная схема представляет собой графическое описание моделируемой системы, сохраняемое в файле с расширением `.sch`. Она включает в себя все компоненты модели, их параметры, соединения, начальные и граничные условия. На основе схемы препроцессор автоматически формирует математическую модель, которая затем передаётся решателю.

Схема строится из базовых объектов — **элементов, портов, связей и узлов**. Подробные определения этих понятий приведены в разделе 2.2 «Минимальный словарь». При сборке схемы необходимо строго соблюдать правило совместимости портов: соединять можно только порты одного типа и с одинаковым количеством степеней свободы (**DOF**). Нарушение этого правила приводит к ошибкам при формировании модели.

Совокупность элементов, портов, связей и узлов определяет структуру модели, которая впоследствии преобразуется в систему обыкновенных дифференциальных уравнений, решаемую вычислительным ядром PRADIS.

2.2.4 Как PRADIS формирует математическую модель

PRADIS реализует объектно-ориентированный подход к моделированию. Пользователь описывает физическую систему, собирая расчётную схему из готовых компонентов и соединяя их через совместимые порты.

Каждый компонент содержит математическое описание своего физического поведения, а каждый порт определяет набор физических величин и степеней свободы (**DOF**), через которые компонент взаимодействует с другими элементами модели.

После запуска расчёта препроцессор автоматически выполняет следующие операции:

- анализирует структуру расчётной схемы;
- проверяет корректность соединений между компонентами;
- объединяет связанные порты в узлы;
- определяет внутреннюю нумерацию степеней свободы;
- формирует математическую модель всей системы на основе уравнений отдельных компонентов и их взаимных связей.

В результате автоматически строится система обыкновенных дифференциальных уравнений, описывающая поведение моделируемой системы. Пользователю не требуется самостоятельно выводить уравнения движения или записывать систему уравнений вручную — все необходимые зависимости формируются автоматически на основе структуры расчётной схемы.

После формирования математической модели вычислительное ядро PRADIS выполняет её численное решение, а результаты расчёта становятся доступны в виде графиков, таблиц, анимации и других средств анализа.

2.3 Подготовка к работе

2.3.1 Состав установленного комплекса

Программный комплекс PRADIS состоит из трёх основных компонентов, каждый из которых отвечает за определённый этап моделирования.

1. **Препроцессор (PRADIS Qucs)** — предназначен для создания и редактирования расчётных схем. В нём пользователь собирает модель из компонентов библиотек, соединяет их между собой, задаёт параметры элементов и настраивает условия расчёта.
2. **Вычислительное ядро (решатель)** — выполняет автоматическое построение математической модели на основе расчётной схемы и производит её численное решение. Пользователь напрямую с решателем не взаимодействует — запуск вычислений осуществляется из препроцессора.
3. **Постпроцессор (Postprocessor)** — предназначен для просмотра, анализа и визуализации результатов моделирования. Позволяет строить графики, просматривать таблицы, выполнять анимацию модели и экспортировать полученные данные.

Работа с PRADIS обычно выполняется в следующей последовательности:

PRADIS Qucs → Решатель → Postprocessor

Сначала пользователь создаёт расчётную схему, затем запускает моделирование, после чего анализирует полученные результаты.

2.3.2 Запуск PRADIS/DINAMA

Запуск компонентов программного комплекса может выполняться несколькими способами.

Препроцессор (PRADIS Qucs)

- двойным щелчком по ярлыку **PRADIS Qucs** на рабочем столе;
- запуском файла **pqrun.bat** из каталога C:\DINAMA\Qucs\bin

После запуска открывается главное окно PRADIS Qucs, в котором выполняется создание проектов, построение расчётных схем, настройка параметров модели и запуск расчёта.

Постпроцессор (Postprocessor)

- автоматически после успешного завершения расчёта;
- вручную с помощью ярлыка **Post**;
- запуском файла **Postprocessor.exe**, расположенного в каталоге C:\DINAMA\post

В постпроцессоре доступны графики, таблицы, анимация модели, экспорт результатов и другие средства анализа.

2.3.3 Проверка доступности библиотек элементов

После запуска PRADIS Qucs рекомендуется убедиться, что библиотеки компонентов загружены корректно.

В **Главном прикрепляемом окне** перейдите на вкладку «**Компоненты**». Доступные библиотеки отображаются **на панели библиотек**, расположенной справа от Главного прикрепляемого окна. После выбора библиотеки её содержимое отображается в **раскрываемом списке**, расположенном под строкой поиска компонентов, а ниже — в виде панели с компонентами выбранной библиотеки.

В списке должны присутствовать, например, следующие библиотеки:

- **Базовые модели;**

- **Сигналы;**
- **Линейная механика;**
- **Гидравлика;**
- **и другие библиотеки.**

Если библиотеки отображаются в панели выбора, а при их выборе становятся доступны соответствующие компоненты, установка программного комплекса выполнена корректно и можно приступить к созданию расчётной схемы.

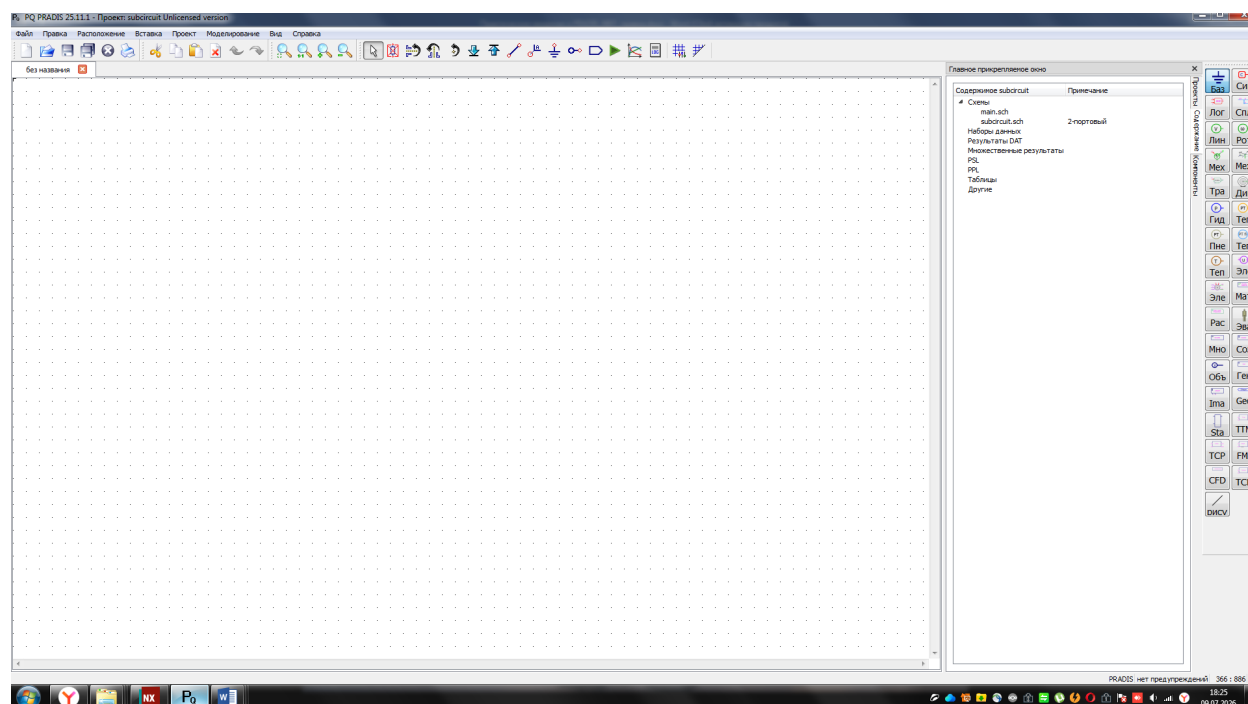


Рис. 3.3 Главное прикрепляемое окно PRADIS Qucs: панель библиотек, раскрывающийся список библиотек и компоненты выбранной библиотеки.

После установки PRADIS пользовательские проекты, примеры и служебные файлы располагаются в разных каталогах.

По умолчанию проекты пользователя хранятся в скрытом каталоге:

C:\Users\<имя_пользователя>\\.pq\

Для каждого проекта создаётся отдельная папка, содержащая все файлы, необходимые для моделирования. Внутри неё обычно находятся:

- **схемы (*.sch);**
- **наборы данных;**
- **результаты расчётов (*.dat);**
- **файлы PSL/PPL (скрипты и задания для решателя);**
- **другие служебные файлы проекта.**

Готовые примеры моделей устанавливаются вместе с программным комплексом и располагаются в каталоге: C:\DINAMA\examples

Этот каталог содержит демонстрационные проекты, предназначенные для изучения возможностей PRADIS и использования в качестве основы при создании собственных моделей.

Важно

Каталог **.pq** является скрытым. Если он не отображается в Проводнике Windows, необходимо включить отображение скрытых файлов и папок в настройках операционной системы.

2.4 Пример: простая механическая схема

В руководстве используется один сквозной пример — пружинный маятник без учета силы тяжести. На нем показан полный путь от идеи модели до графиков результатов. В процессе работы мы проделаем следующие шаги:

1. Создадим проект PRADIS.
2. Создадим файл расчетной схемы.
3. Выберем библиотеки и найдем элементы модели.
4. Разместим массу, пружину, демпфер, опору и источник начального смещения.
5. Соединим элементы через совместимые порты.
6. Зададим параметры элементов и расчета.
7. Запустим расчет.
8. Откроем результаты в постпроцессоре.
9. Проверим физический смысл графиков.

2.4.1 Постановка задачи

Построить модель пружинного маятника и проанализировать зависимость скорости и перемещения от времени при свободных колебаниях. Сила тяжести не учитывается.

Исходные данные: Жёсткость пружины $k=1$ Н/м, Вязкость $\mu=0,2$ Н*с/м, Масса $m=1$ кг, Начальное смещение $X_0=2$ м.

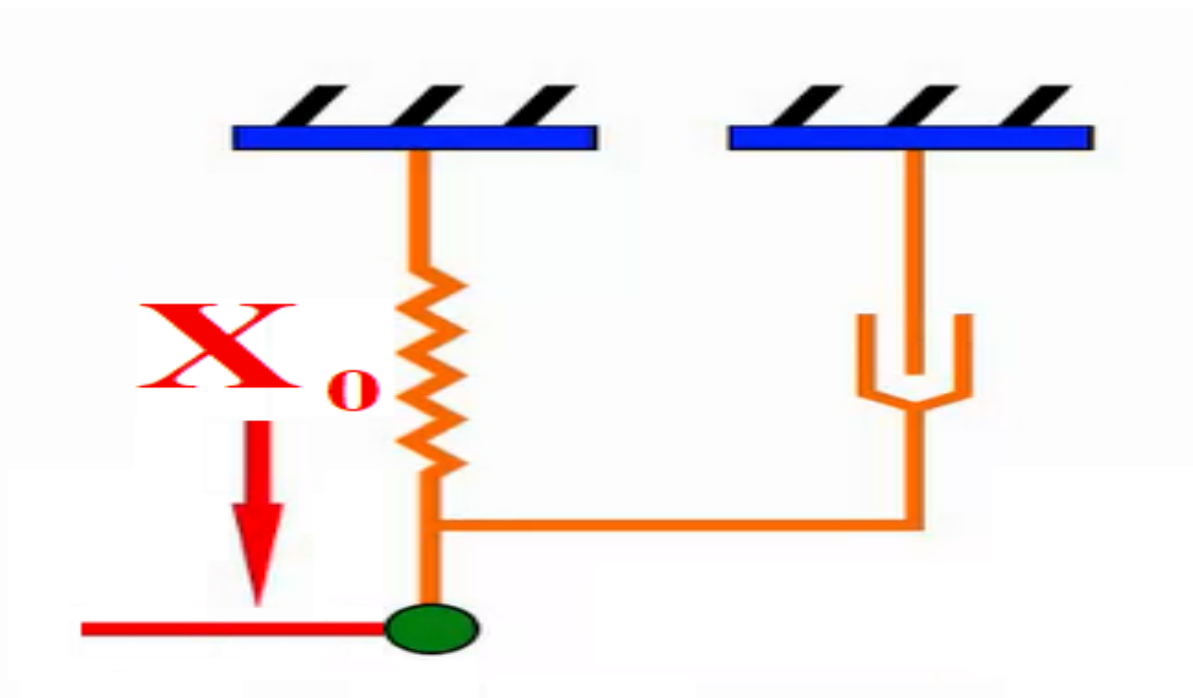


Рис. 4.1 Модель пружинного маятника

2.4.2 Создание проекта

1. Для построения расчетной схемы/модели пружинного маятника запустить препроцессор. Запуск осуществляется двойным щелчком мыши по ярлыку **PRADIS Qucs** на рабочем столе

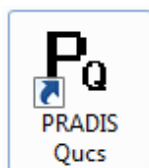


Рис. 4.2.1 Ярлык PRADIS Qucs

Откроется окно препроцессора

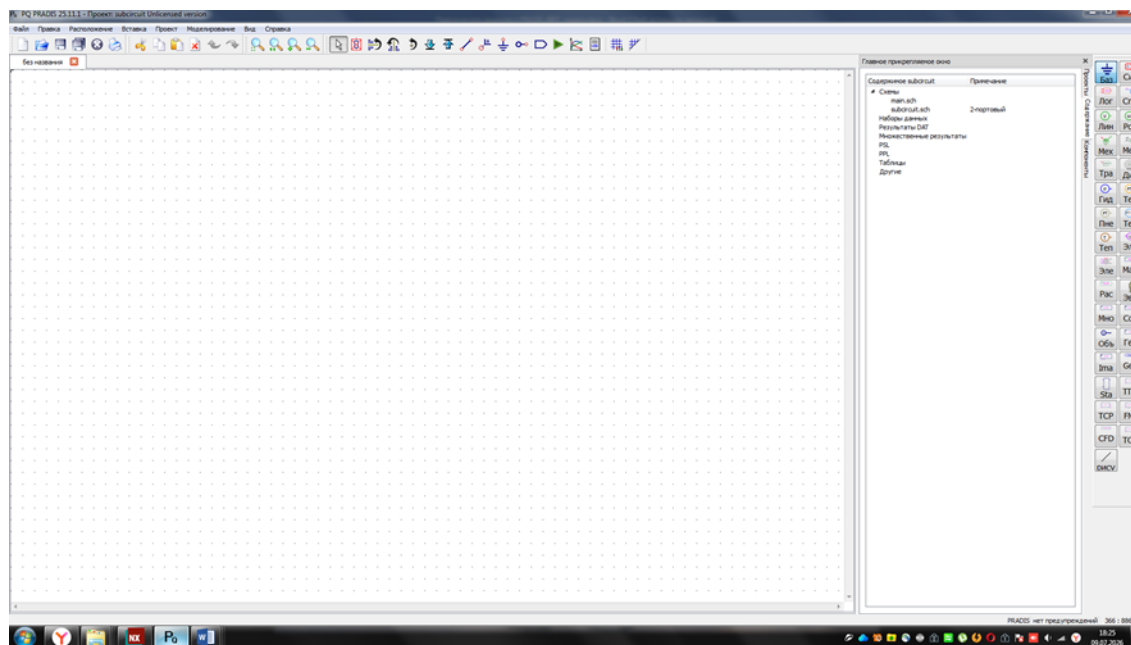


Рис. 4.2.2 Главное окно препроцессора после запуска.

2. На панели управления проектом нажать на кнопку «Создать». Откроется окно диалога для определения имени проекта

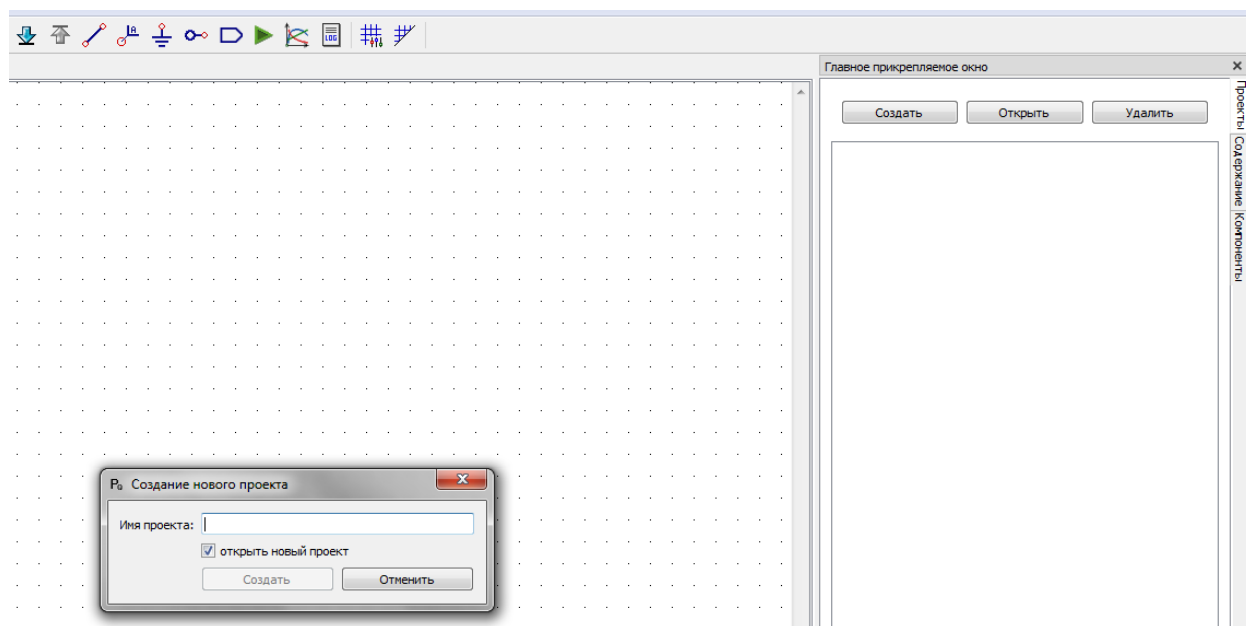


Рис. 4.2.3 Создание нового проекта

3. Пользователю необходимо задать имя проекта (например, Project1), после чего произойдет активация кнопки «Создать». Оставить, либо убрать галочку «Открыть новый проект». При установленной галочке проект открывается автоматически. Нажать на кнопку «Создать». В Главном прикрепляемом окне появится проект Project1. Нажав на имя проекта два раза, откроется структура проекта. Сам проект, по умолчанию, создается для ОС Windows в каталоге: C:\Users\<имя пользователя>\\.pq

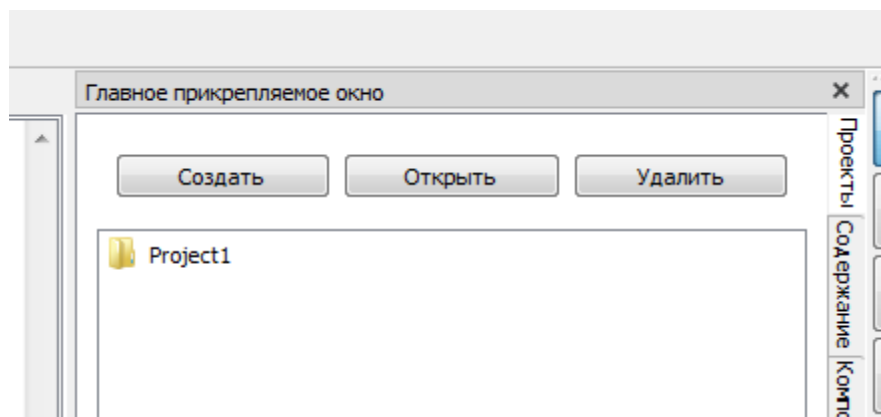


Рис. 4.2.4 Структура главного прикрепляемого окна

4. По своей сути, проект представляет собой папку, содержащую все данные и файлы, относящиеся к создаваемой модели. Для построения модели нужно создать файл с расширением «.sch», который будет содержать код, описывающий схему модели. Перейти на вкладку «Файл→Сохранить как...». Откроется стандартное диалоговое окно ОС Windows для сохранения файла. Проверить путь на каталог: C:\\Users\\<имя пользователя>\\.pq\\ Project1. В поле «Имя файла:» задать имя файла pendulum, расширение «.sch» добавится автоматически.

Важно

В пути к файлу не должно быть знаков кириллицы и пробелов.

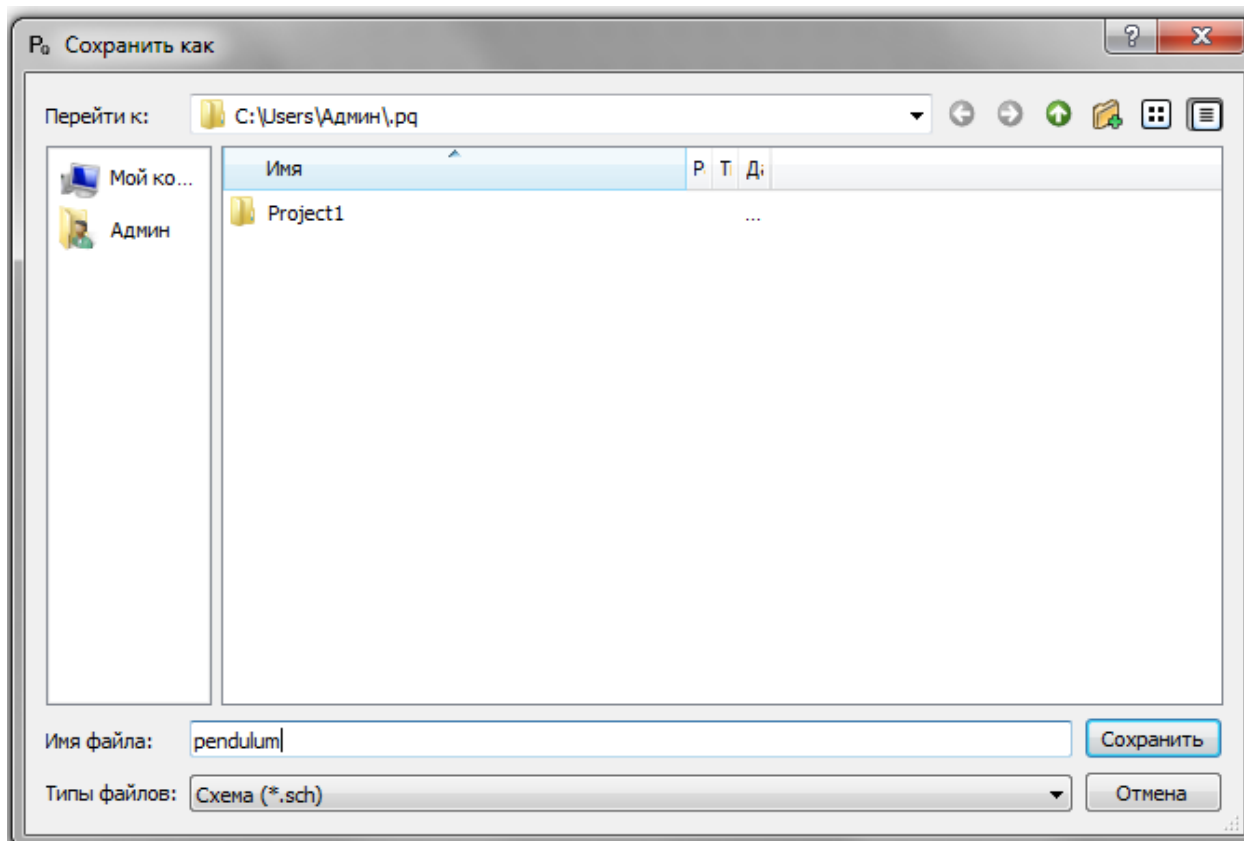


Рис. 4.2.5 Сохранение файла

5. Нажать кнопку «Сохранить». В результате название файла «pendulum.sch» появится в левом верхнем углу рабочего поля.

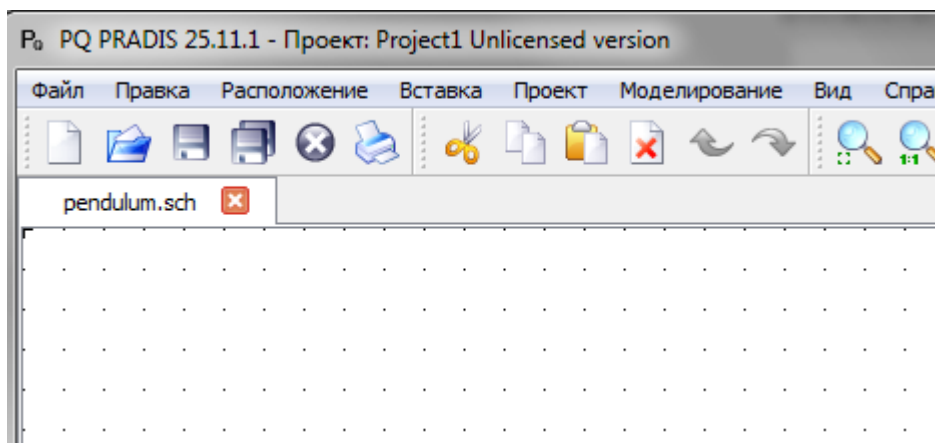


Рис. 4.2.6 Отображение созданного файла

2.4.3 Состав схемы

Для построения модели необходимо:

1. определить элементы, из которых она состоит;
2. определить, в какой библиотеке находятся эти элементы;

3. разместить элементы модели на схеме с учетом степеней свободы.

Модель пружинного маятника состоит из пружины, груза (массы), демпфера (элемента вязкости) и крепления. Его движение возникает после начального возмущения – отклонения груза на величину X_0 от положения равновесия.

При определении библиотеки, содержащей элементы пружинного маятника, необходимо руководствоваться физической природой модели, либо составных частей модели. В данном случае, пружинный маятник относится к разделу одномерной механики, поскольку её динамика полностью определяется **одной степенью свободы**. Это означает, что для описания состояния системы (положения и скорости груза) в любой момент времени достаточно одной обобщённой координаты, например, деформации пружины x . Уравнение движения, выведенное из второго закона Ньютона $m \times d^2x/dt^2 = -k \times x$, является **обыкновенным дифференциальным уравнением второго порядка относительно одной переменной $x(t)$** , что является классическим признаком одномерной механической системы.

Выбираем необходимые компоненты из библиотек для создания расчетной схемы.

Из библиотеки «**Механика 1D**»:

- «**Земля**» (Ground) — абсолютная опора;
- **K** — упругая связь (пружина);
- **MU** — элемент вязкости (демпфер);
- **M** — инерционный элемент (масса);
- **X0** — начальное перемещение (источник начального смещения).

Из библиотеки «**Базовые компоненты**»:

- индикаторы **X** (перемещение) и **V** (скорость);
- компоненты управления расчётом **Dynamic** и вывода результатов **DISP**.

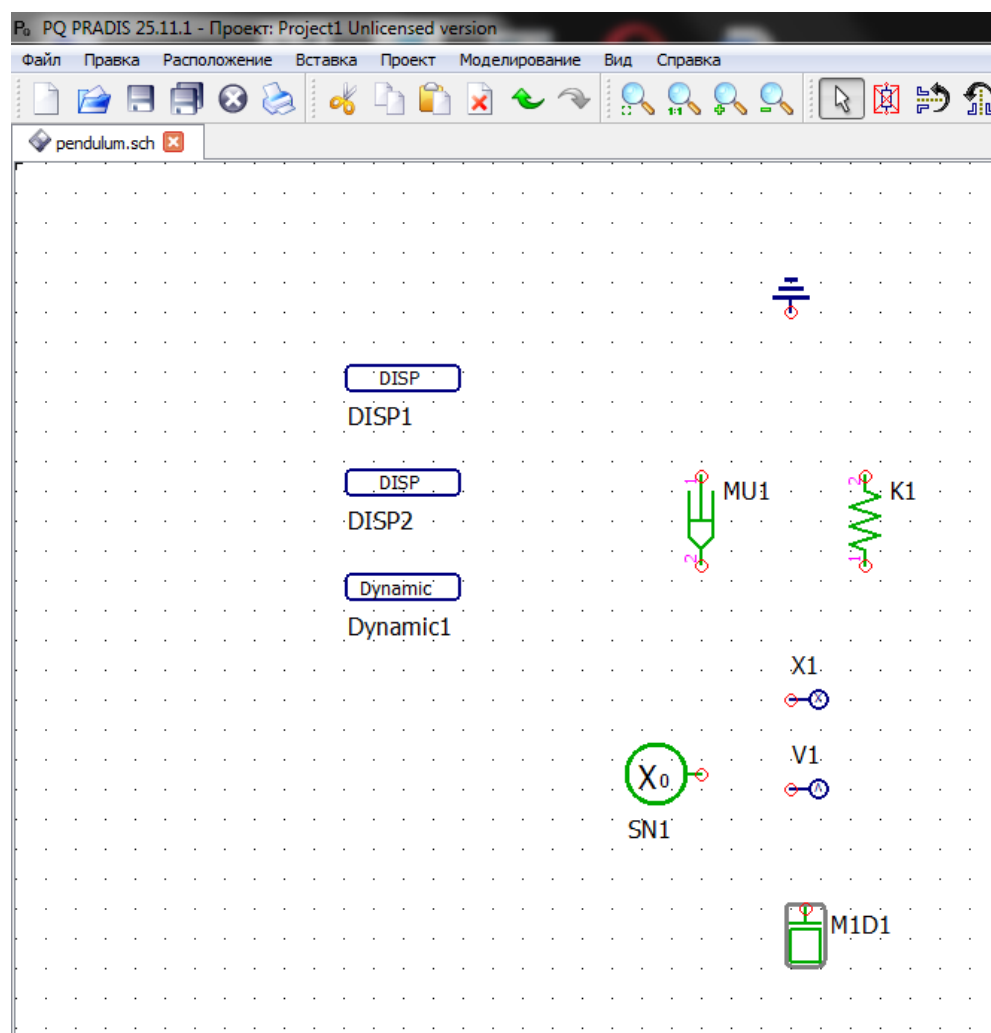


Рис. 4.3.1 Расположение элементов на схеме

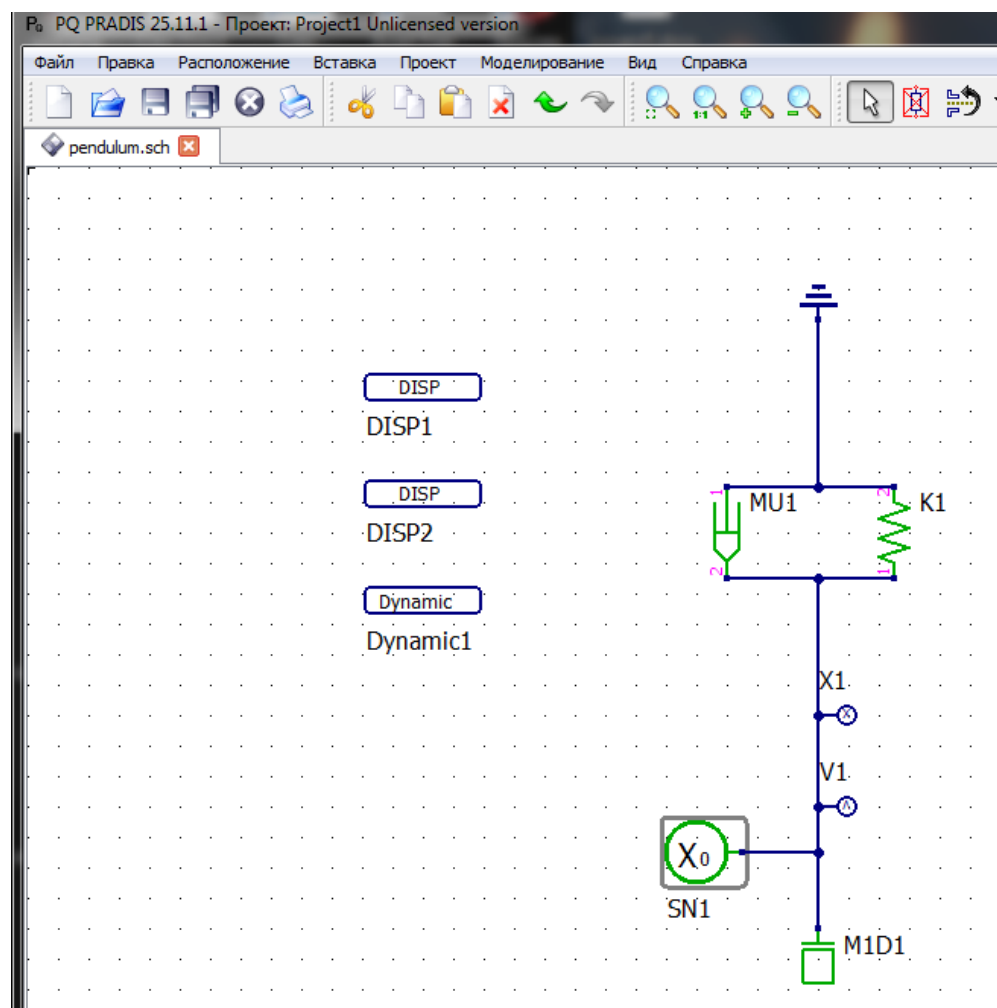


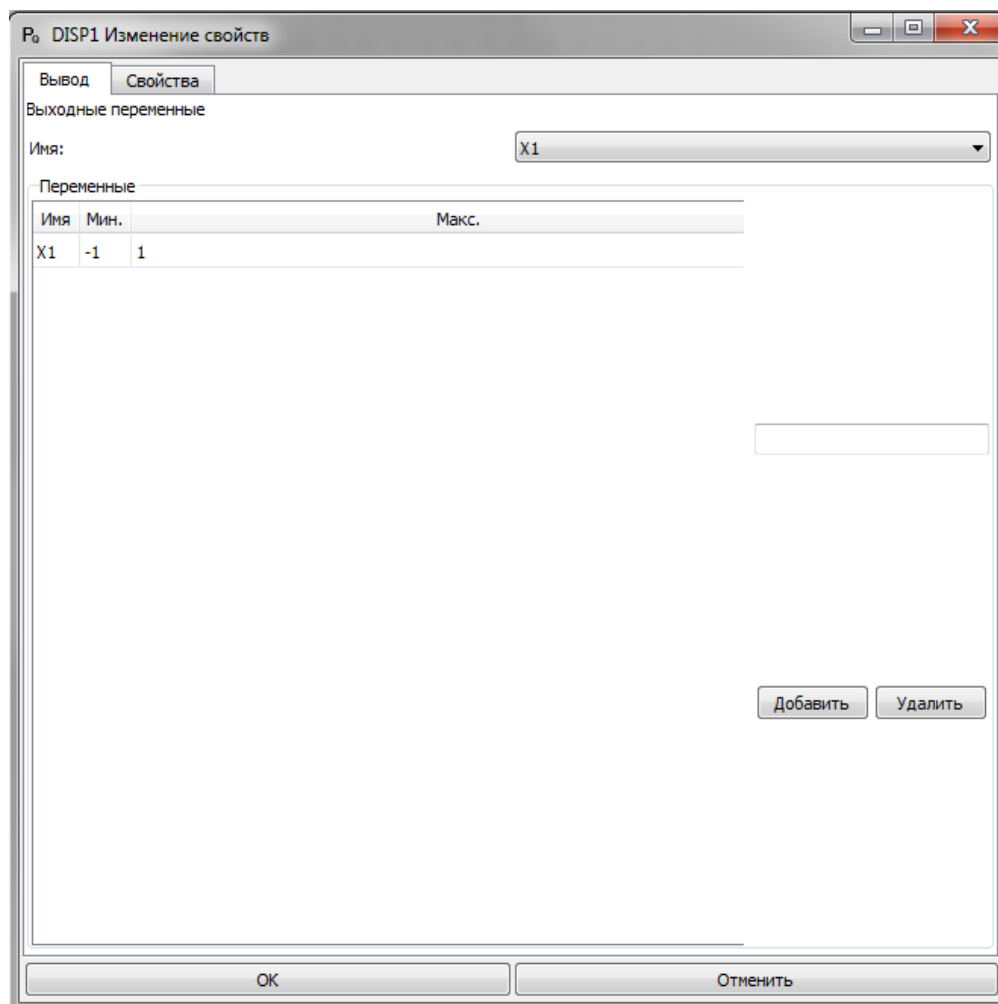
Рис. 4.3.2 Соединение элементов на схеме при помощи функции «Проводник»

2.4.4 Настройка параметров

Для каждого элемента необходимо задать физические параметры. Двойным щелчком левой кнопки мыши по элементу открывается окно свойств, в котором следует указать:

- **K**: жёсткость = 1 (Н/м);
- **MU**: вязкость = 0,2 (Н·с/м);
- **M**: масса = 1 (кг);
- **X0**: смещение = 2 (м);
- **Dynamic**: конечное время расчёта end = 15 (с).

В компоненте **DISP** необходимо задать ссылки на датчики для вывода результатов. В свойствах DISP в соответствующем поле указываются идентификаторы индикаторов X1 и V1 (см. Рис. 4.3.2).



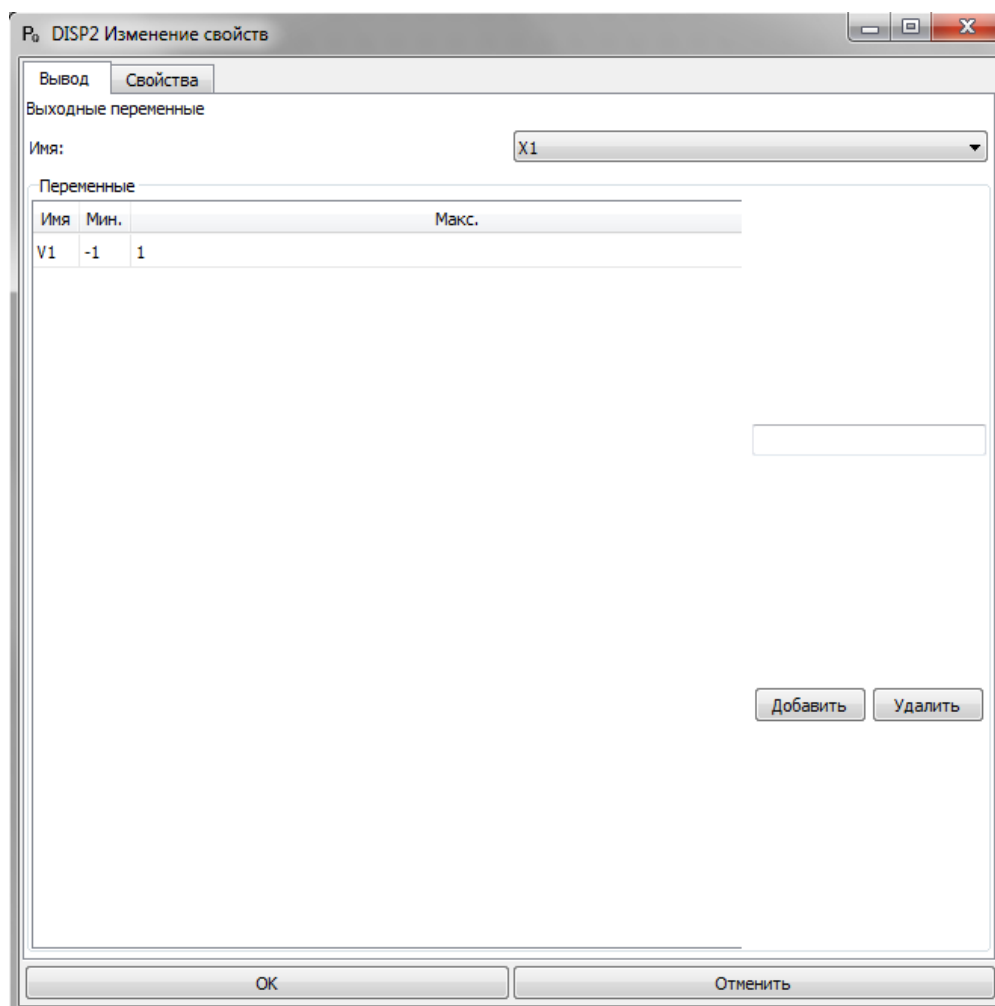


Рис. 4.3.2 Задание ссылки на датчики X1 и V1 для выходных параметров DISP

2.4.5 Запуск расчёта

Запуск расчёта выполняется нажатием клавиши **F2** или выбором команды **Моделирование** → **Моделировать**. В консоли отображается процесс генерации PSL/PPL-файлов и ход численного интегрирования. После успешного завершения расчёта автоматически открывается окно постпроцессора.

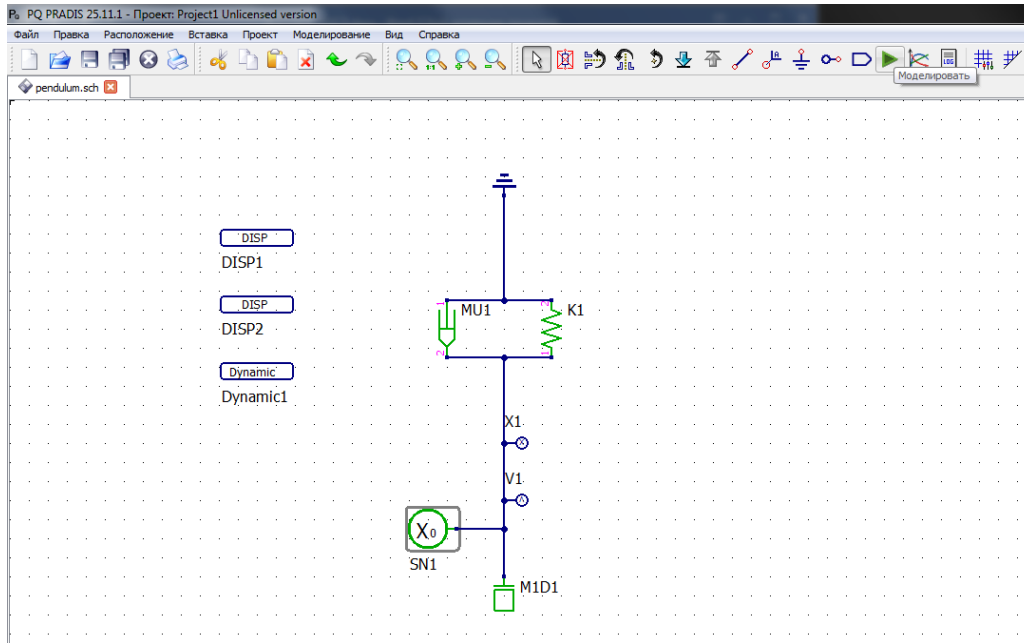


Рис. 4.5.1 Запуск расчета схемы .. image:: media/image17.png

width

5.35451in

height

3.32869in

Рис. 4.5.2 Процесс расчёта схемы: вывод статусных сообщений в консоль (условное изображение).

2.4.6 Анализ результата

После завершения моделирования автоматически открывается окно **PRADIS Postprocessor** с графиками, указанными в компоненте DISP.

На графиках должны отображаться:

- перемещение (от индикатора X) в зависимости от времени;
- скорость (от индикатора V) в зависимости от времени.

Ожидаемый вид графиков — затухающие колебания:

- в начальный момент система имеет максимальное смещение ($X=2$ м);
- затем масса начинает колебаться с постепенным уменьшением амплитуды;
- из-за наличия элемента вязкости (MU) амплитуда экспоненциально затухает.

Такое поведение свидетельствует о корректной работе модели и успешном выполнении расчёта решателем.

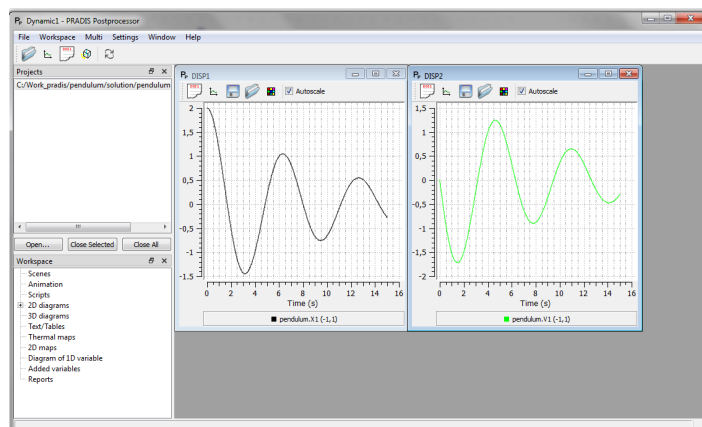


Рис. 4.5.3 Графики перемещения и скорости.

В результате работы программы пользователь получает:

- модель;
- расчёт;
- график перемещения;
- график скорости.

2.5 Алгоритм по созданию расчетной схемы.

Сборка расчётной схемы представляет собой процесс создания графической топологии моделируемой физической системы. На данном этапе пользователь определяет структуру модели, выбирает необходимые компоненты из библиотек препроцессора и устанавливает связи между ними в соответствии с физическими взаимодействиями. Корректность сборки является обязательным условием для автоматического формирования математической модели и успешного выполнения численного расчёта; любые ошибки на этапе сборки приводят к сбоям при генерации модели или в ходе решения.

Процесс сборки включает следующие последовательные операции:

- выбор библиотеки компонентов, соответствующей физическому домену;
- размещение выбранных элементов на рабочем поле;
- соединение портов элементов проводниками;
- контроль точности привязки к координатной сетке;
- проверка целостности и замкнутости топологии схемы.

2.5.1 Выбор библиотеки элементов

Выбор библиотеки определяется физической природой моделируемой системы. В препроцессоре PRADIS каждая библиотека содержит компоненты, специализированные для определённого домена.

Для выбора библиотеки пользователю необходимо выполнить следующие действия:

1. В **Главном прикрепляемом окне** перейти на вкладку «**Компоненты**».
2. На панели библиотек выбрать библиотеку, соответствующую решаемой задаче:

- **Линейная механика** — для систем с поступательным движением (перемещения, силы, массы, упругие и диссипативные элементы);
- **Роторная механика** — для систем с вращательным движением (углы поворота, моменты, инерционные элементы);
- **Гидравлика / Пневматика** — для моделирования систем с рабочими жидкостями или газами под давлением (насосы, клапаны, трубопроводы);
- **Сигналы** — для построения систем управления, обработки сигналов и логических цепей (источники сигналов, сумматоры, регуляторы, преобразователи).

3. После выбора библиотеки в раскрывающемся списке становятся доступны её тематические разделы, а на панели ниже отображаются пиктограммы компонентов, предназначенных для использования в схеме.

Примечание. Для любой расчётной схемы, помимо профильных физических элементов, требуется установка вспомогательных компонентов из библиотек «**Базовые компоненты**» и «**Сигналы**». К числу обязательных относятся:

- **Ground** («Земля») — элемент абсолютной опоры, задающий неподвижную систему отсчёта в механических системах;
- **Dynamic** — компонент управления динамическим расчётом, определяющий временные параметры моделирования;
- **DISP** — компонент управления выводом результатов в постпроцессор;
- индикаторы **X** и **V** — для визуализации значений переменных (перемещений, скоростей, сил, давлений и пр.) непосредственно на схеме.

Отсутствие указанных элементов делает невозможным корректное формирование математической модели и выполнение расчёта.

2.5.2 Размещение элементов на схеме

После определения необходимого набора компонентов пользователь приступает к их размещению на рабочем поле. Правильное расположение элементов улучшает наглядность модели и облегчает последующую прокладку соединений.

Для размещения элемента следует выполнить указанные ниже действия.

1. В панели компонентов найти пиктограмму требуемого элемента.
2. Перетащить компонент левой кнопкой мыши на рабочую область схемы. Альтернативный метод: щёлкнуть по компоненту левой кнопкой — элемент будет зафиксирован на курсоре, после чего необходимо переместить его в желаемую позицию и щёлкнуть повторно для фиксации.
3. Повторить операцию для всех элементов, входящих в состав модели.
4. При необходимости изменить ориентацию элемента (поворот на 90° по часовой стрелке, 90° против часовой стрелки или на 180°). Для этого следует использовать **правую кнопку мыши** — вызвать контекстное меню и выбрать пункт «Повернуть» (или применить соответствующую комбинацию клавиш, если она задана в настройках). Поворот допустим как на этапе, когда элемент ещё зафиксирован на курсоре (до фиксации), так и после его размещения (предварительно выделив компонент).

Размещение элементов следует выполнять в соответствии с физической структурой моделируемой системы: источники воздействия — слева, исполнительные механизмы —

справа; механические цепи — горизонтально, гидравлические тракты — по направлению потоков среды. Между компонентами необходимо оставлять интервал не менее 2–3 шагов координатной сетки для обеспечения удобства прокладки проводников. При наличии параллельных или последовательных ветвей рекомендуется группировать их визуально для повышения читаемости схемы и снижения вероятности ошибок при соединении.

2.5.3 Соединение портов

Важно

Взаимодействие элементов в PRADIS реализуется через порты. Установление соединения между портами означает создание физической связи, через которую передаются соответствующие физические величины (силы, моменты, давления, расходы, электрические сигналы и др.) в соответствии с типом библиотеки и моделью компонента.

Для создания проводниковых соединений пользователю необходимо выполнить следующую последовательность действий:

1. Выбрать команду **«Вставка → Проводник»** или нажать сочетание клавиш **Ctrl + E**.
2. Подвести указатель мыши к порту первого элемента (порт визуально обозначен кружком или прямоугольником на границе компонента) и щёлкнуть левой кнопкой.
3. Провести линию к порту второго элемента и снова щёлкнуть левой кнопкой для завершения соединения.
4. Повторить шаги 2–3 для всех необходимых связей.

Важное правило совместимости портов. Соединение допустимо только между портами одного типа и с одинаковым количеством степеней свободы (DOF). При попытке соединить несовместимые порты программа выдаст сообщение об ошибке.

2.5.4 Правила привязки к сетке и размещения портов

При соединении портов (см. п. 2.2) необходимо соблюдать следующие правила.

- **Точность попадания в порты.** Проводник должен начинаться и заканчиваться точно в геометрическом центре порта. Любое смещение, даже на долю шага сетки, считается разрывом соединения и приводит к ошибкам при формировании системы уравнений.
- **Управление отображением сетки.** Для визуального контроля привязки можно включить отображение сетки через соответствующую иконку на панели инструментов или через меню **Файл → Настройки документа**. Однако шаг сетки (по умолчанию 10 единиц) изменять запрещено.
- **Коррекция смещённых элементов.** Если в процессе редактирования элементы были смещены относительно узлов сетки, необходимо выделить их и выполнить команду **Расположение → Выравнивать по сетке**. Это восстановит правильное позиционирование.
- **Недопустимость неподключённых портов.** Все порты, которые по физической модели должны быть задействованы, обязаны иметь соединения. Неподключённые порты (если их свободное состояние не предусмотрено документацией компонента)

являются ошибкой, приводящей к сингулярности матрицы и аварийному завершению расчёта.

2.5.5 Проверка корректности соединений

Перед переходом к настройке параметров элементов и запуску расчёта необходимо выполнить обязательную проверку схемы на предмет целостности, замкнутости и отсутствия дефектов топологии. Проверка осуществляется как визуально, так и с использованием встроенных средств препроцессора.

Рекомендуется следующий порядок проверочных действий.

1. **Масштабирование для визуального контроля.** Использовать команды Вид → Показать все (отображает всю схему в пределах рабочей области) и Вид → Масштаб 1:1 (устанавливает реальный масштаб). При масштабе 1:1 следует тщательно осмотреть каждый проводник на предмет точного касания портов. При обнаружении зазоров (даже минимальных) соединение необходимо пересоздать.
2. **Проверка замкнутости физических контуров.** Все физические цепи должны быть замкнуты. В механических системах каждая кинематическая цепь должна иметь опору на абсолютную систему отсчёта, представленную компонентом **Ground**. В гидравлических и пневматических системах должны быть определены граничные условия (например, соединение с атмосферой или источником давления). Незамкнутые цепи приводят к ошибкам при формировании уравнений движения.
3. **Контроль отсутствия лишних пересечений и случайных узлов.** Следует визуально проверить, что проводники не пересекаются в непредусмотренных местах, так как каждое пересечение автоматически создаёт узел, что может исказить топологию модели. При необходимости следует перепроложить проводники, избегая нежелательных пересечений.
4. **Проверка наличия обязательных вспомогательных компонентов.** Убедиться, что на схеме присутствуют **Dynamic**, **DISP**, а также необходимые индикаторы и элементы заземления/опоры. При отсутствии любого из них расчёт не будет запущен.
5. **Использование автоматической проверки (если доступна).** В некоторых версиях препроцессора предусмотрена команда автоматической верификации схемы (например, через меню Моделирование → Проверить схему или аналогичную). Рекомендуется применять эту функцию для выявления скрытых ошибок, неочевидных при визуальном контроле.

Схема считается готовой к дальнейшей настройке параметров и начальных условий только после успешного прохождения всех указанных проверок. Наличие любой из перечисленных ошибок делает невозможным корректное формирование математической модели и выполнение расчёта.

2.6 Контрольный список первого самостоятельного расчёта

Данный раздел содержит перечень обязательных проверок, которые рекомендуется выполнять перед запуском расчёта и после его завершения для обеспечения достоверности результатов.

2.6.1 Перед запуском расчёта

- Проект создан и сохранён.
- Схема содержит все необходимые элементы.
- Элементы соединены через совместимые порты.
- Параметры элементов заданы в корректных единицах измерения.
- Начальные условия (смещения, скорости и др.) соответствуют физической постановке задачи.
- Внешние воздействия (силы, давления, сигналы) заданы явным образом.
- Выбран корректный тип расчёта (динамический анализ).
- В компоненте Dynamic заданы разумные значения конечного времени и шага вывода.
- Схема не содержит незамкнутых физических цепей (например, механическая система имеет опору на Ground).
- Результаты можно предварительно оценить приблизительной инженерной оценкой.

2.6.2 После завершения расчёта

- Расчёт завершился без сообщений об ошибках (консоль не содержит критических предупреждений).
- Ключевые переменные (перемещения, скорости, силы, давления) доступны для просмотра в постпроцессоре.
- Графики не содержат неестественных скачков, разрывов или расходимости.
- Направления перемещений, сил или потоков соответствуют ожидаемым на основе постановки задачи.
- Изменение основных параметров (массы, жёсткости, вязкости) приводит к ожидаемому изменению результатов (проверка чувствительности).

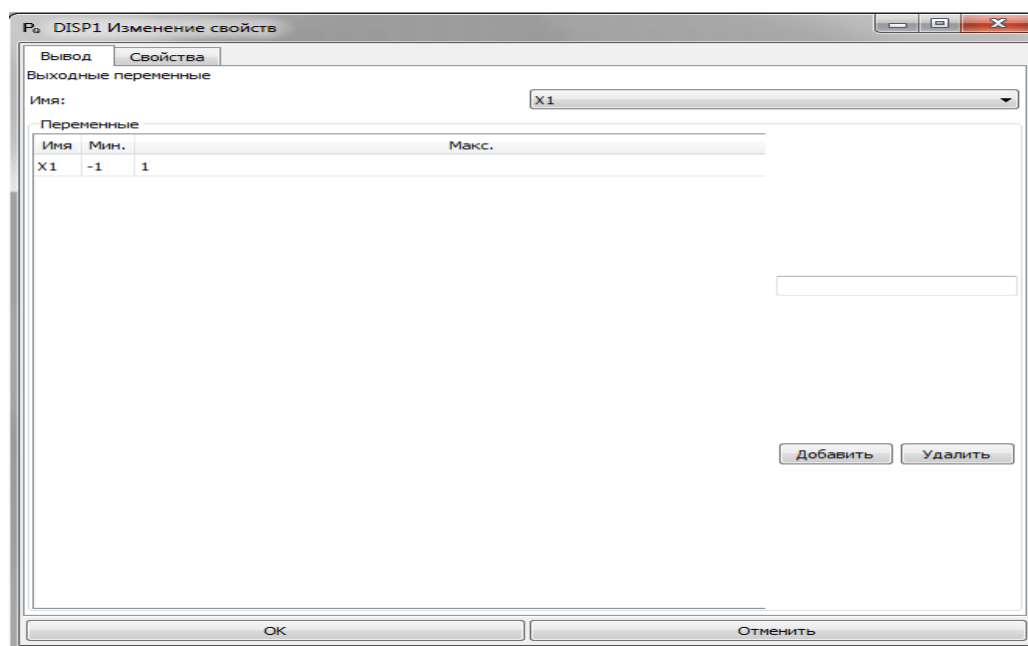
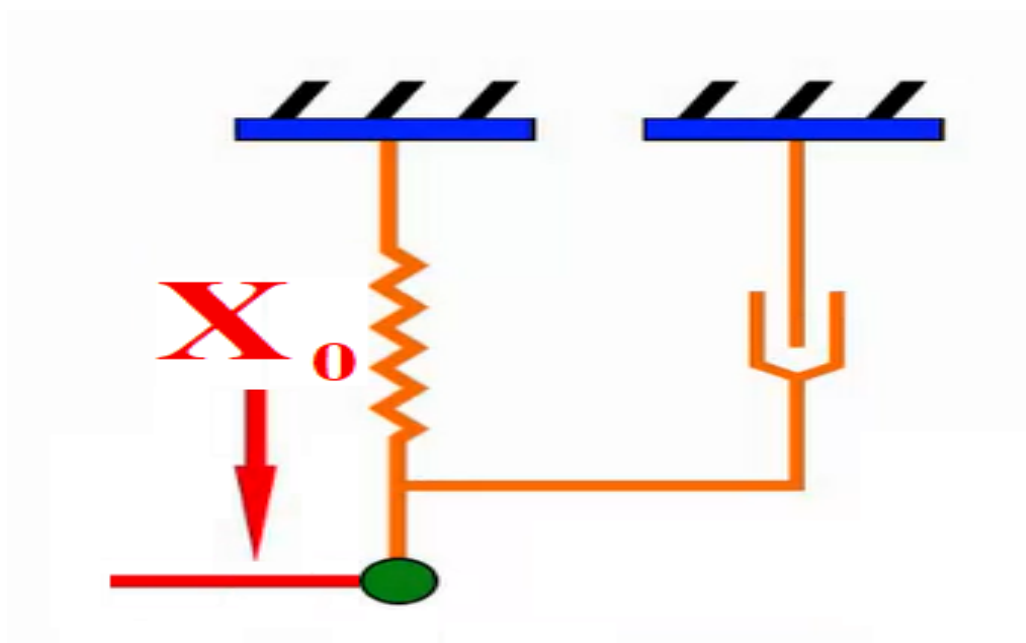
2.7 Типичные ошибки

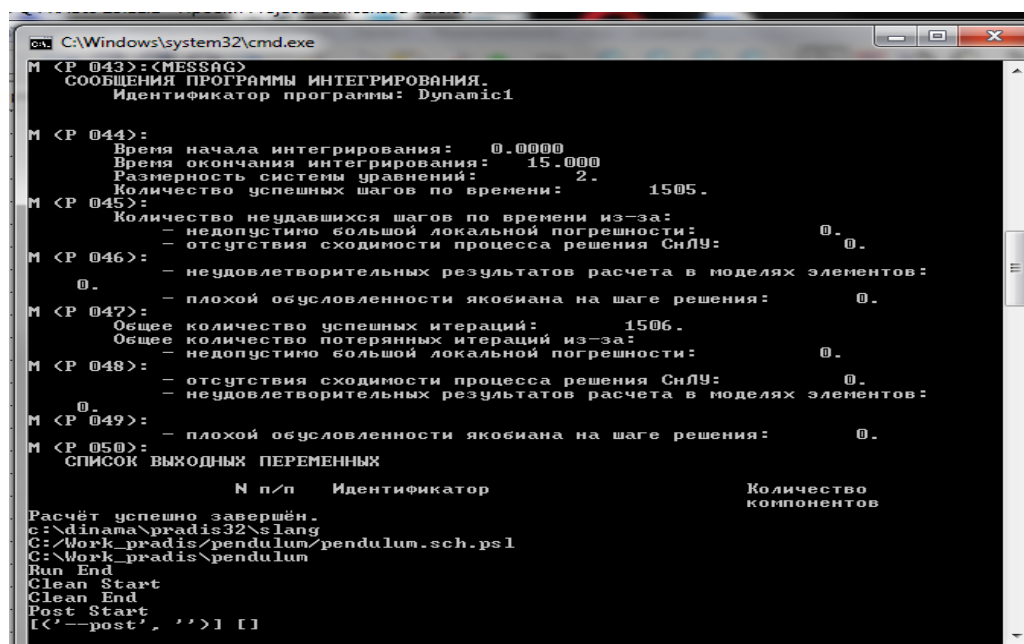
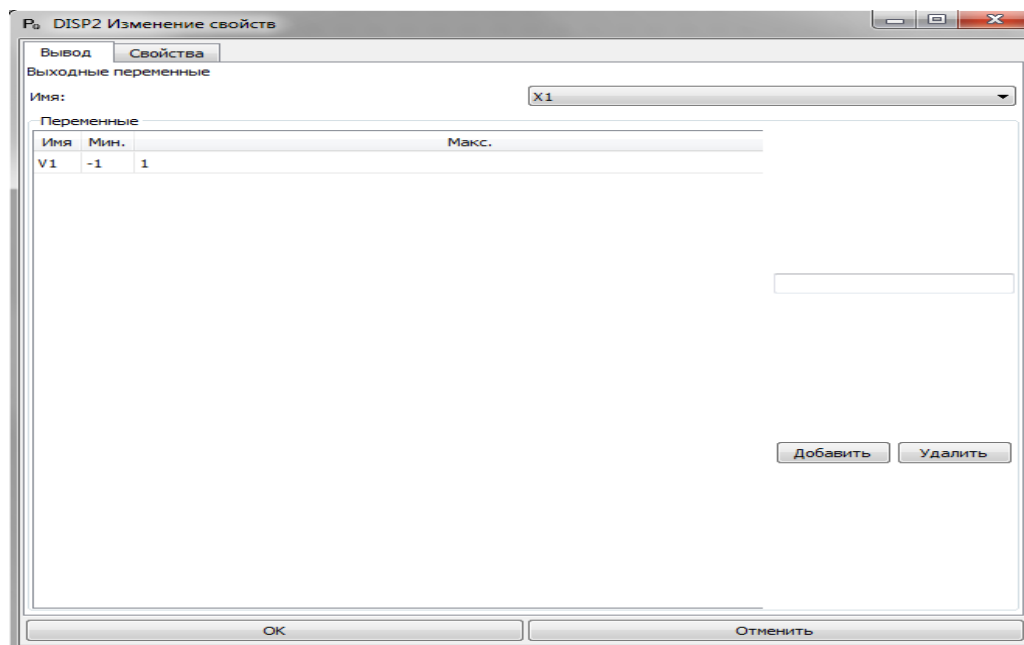
Симптом (ситуация)	Что проверить / Возможная причина
Элемент не найден в библиотеке.	Проверить список доступных библиотек, пути к ним, наличие элемента в справочнике, корректность установки комплекса.
Не удаётся соединить порты.	Проверить тип каждого порта, их расположение на схеме, привязку к сетке (шаг 10), корректность выбранного физического домена.
Расчёт не запускается.	Проверить наличие компонентов Dynamic, Ground, DISP; прочитайте диагностические сообщения в консоли; убедиться в отсутствии неподключённых портов; проверить доступность библиотек.
Ошибка сингулярности матрицы.	Отсутствие элемента Ground (абсолютной системы отсчёта) в механической схеме.
Расчёт расходится (численная неустойчивость).	Проверить начальные значения, параметры расчёта (шаг, конечное время), величины нагрузок, единицы измерения, наличие чрезмерно больших или нулевых коэффициентов (например, нулевая масса или жёсткость).
Нет графиков / результатов в пост-процессоре.	Не добавлены индикаторы (X, V) или компонент DISP; в DISP не указаны ссылки на индикаторы.
Результат выглядит неправдоподобно (физически некорректно).	Пересмотреть физический смысл схемы, правильность направления осей, значения параметров, ожидаемый порядок величин; проверить простой предельный случай (например, при нулевом демпфировании колебания не должны затухать).

2.8 Заключение

Практическая Практическая работа в PRADIS строится вокруг расчётной схемы. Пользователь выбирает элементы из библиотек, соединяет их через порты, задаёт параметры и условия расчёта, после чего комплекс автоматически формирует и решает математическую модель. На первом этапе важно освоить не все возможности PRADIS, а базовый рабочий цикл: схема, параметры, расчёт, результат, проверка. После этого пользователь сможет осмысленно переходить к более сложным темам: подсхемам, объектам, пользовательским моделям, расширению библиотек и детальному анализу математической основы комплекса.

Успешное освоение материала позволяет пользователю самостоятельно моделировать динамические системы различной физической природы и получать достоверные результаты, что открывает возможности для решения широкого круга инженерных и научных задач.





Руководство пользователя PRADIS

3.1 Руководство по работе с препроцессором

3.1.1 Введение и назначение

Препроцессор PRADIS — это интерактивная графическая среда, предназначенная для формирования, параметризации и подготовки расчетных моделей технических систем различной физической природы. Главная задача препроцессора заключается в преобразовании интуитивно понятного графического представления системы (схемы) в строгий машиночитаемый формат, готовый для передачи вычислительному решателю.

Настоящее руководство определяет рекомендуемую последовательность действий (методику) работы с препроцессором при построении динамических моделей. Документ описывает сквозной цикл моделирования: от инициализации проекта до анализа результатов, опираясь исключительно на штатный функционал программного комплекса.

3.1.2 Архитектура и организация рабочего места

Препроцессор интегрирует в единое окно инструментальные панели, векторный редактор схем, систему управления проектами и каталоги готовых библиотек компонентов.

Структура проекта

Проект представляет собой изолированный каталог, содержащий все данные, относящиеся к создаваемой модели. По умолчанию для ОС Windows проект размещается по пути:

C:\Users<имя пользователя>\.qucs<имя проекта>

Внутренняя организационная структура проекта включает папки:

- Схемы - файлы описания схем (расширение .sch);
- Наборы данных - исходные и промежуточные параметры;

- Результаты DAT - файлы расчетных данных;
- PSL / PPL - сгенерированные задания для решателя и скрипты на языке Python;
- Таблицы и другие - вспомогательные данные и паспорта моделей.

Основные элементы интерфейса

- **Главное прикрепляемое окно:** содержит вкладки «Проекты» (управление жизненным циклом файлов) и «Компоненты» (поиск и выбор элементов из библиотек).
- **Панель компонентов:** быстрый доступ к перечню доступных физических доменов (механика, гидравлика, пневматика, автоматика, электрические цепи и др.).
- **Рабочая область:** векторное поле с разметкой-сеткой для построения и выравнивания элементов схемы.
- **Меню и Панель инструментов:** стандартные команды файлового управления, редактирования, вставки, настройки вида и запуска расчетов.

3.1.3 Пошаговая методика создания и анализа модели

Инициализация проекта и схемы

1. Запустите препроцессор через ярлык PRADIS Qucs или файл pqrn.bat.
2. Во вкладке «Проекты» нажмите кнопку **«Создать»**, задайте имя проекта и активируйте опцию «Открыть новый проект».
3. Создайте файл расчетной схемы: перейдите в Файл → Сохранить как..., укажите путь к каталогу проекта, задайте имя файла (расширение .sch добавится автоматически).

Графическое моделирование («Схемотехника»)

1. **Выбор библиотеки:** Определите физическую природу моделируемого объекта и выберите соответствующую библиотеку во вкладке «Компоненты» (например, «Механика 1D» для систем с одной степенью свободы).
2. **Размещение элементов:** Перетащите необходимые компоненты на рабочую область. Используйте инструменты меню Расположение для выравнивания по сетке, центру или краям.
3. **Физическое соединение:** Активируйте инструмент Вставка → Проводник (или Ctrl+E) и соедините элементы через специальные точки — **«потенциалы»** или **«порты»**. Соединение портов означает установление физического взаимодействия (передача усилия, потока, сигнала).
4. При необходимости создания типовых узлов используйте функции Правка → Войти в подсхему / Выйти.

Параметризация и задание начальных состояний

1. Откройте свойства компонента двойным щелчком мыши или через контекстное меню Изменить свойства.
2. Во вкладке **«Свойства»** задайте численные значения:
 - Геометрические характеристики (размеры, сечения, объёмы);

- Физические константы (массы, моменты инерции, коэффициенты сопротивления, жесткости);
 - Эксплуатационные и режимные параметры.
3. Для элементов, обладающих инерционностью или емкостными свойствами, задайте **начальные условия** (начальные координаты, скорости, давления, положения коммутирующих устройств). Корректная инициализация критически важна для адекватного старта динамического расчета.

Определение внешних воздействий и индикаторов

1. Для моделирования нагрузок или управлений используйте компоненты внешних воздействий, поддерживающие константы, ступенчатые, линейные, синусоидальные и произвольные временные зависимости.
2. Добавьте на схему индикаторы из библиотеки «Базовые компоненты»:
 - X - индикатор перемещения или универсальный измеритель внутренних переменных;
 - V - индикатор скорости.
3. Для измерения потоковых или внутренних переменных конкретного компонента используйте синтаксис $N.I(C)$ во вкладке «Выходные значения» свойств индикатора, где:
 - N - имя компонента;
 - I - функция измерения потоковой переменной;
 - C - номер степени свободы. Пример: K1.I(1) для измерения силы упругости в элементе K1.

Формирование задания на расчет

Для корректного запуска моделирования на схеме **обязательно** должны присутствовать два компонента из библиотеки «Базовые компоненты»:

- **Dynamic** - управляет настройками динамического анализа. В свойствах задаются: время расчета (end), метод интегрирования, шаг, точность, интервал вывода в консоль (prtime). Рекомендуется активировать флажок «показывать на схеме» для визуального контроля.
- **DISP** - определяет состав и формат вывода результатов (графики, таблицы) для параметров, отмеченных индикаторами.

Валидация и запуск моделирования

1. Препроцессор автоматически выполняет проверку целостности модели: отсутствие «висящих» портов, согласованность типов данных, наличие обязательных компонентов Dynamic и DISP.
2. Запустите расчет: Моделирование → Моделировать или клавиша F2.
3. Система автоматически:
 - Генерирует скрипт на языке Python (**PPL**);
 - Формирует текстовое задание для динамического решателя (**PSL**);
 - Запускает вычислительное ядро.

4. Ход расчета отображается в консоли. Логи сохраняются в файле **SYSPRINT.txt**.

Анализ результатов в постпроцессоре

По завершении расчета постпроцессор запускается автоматически. Доступные методы анализа:

- **2D-графики:** построение зависимостей выходных переменных от времени с настройкой масштабов, цветов и толщины линий.
- **Фазовые портреты:** визуализация траекторий в координатах «координата – скорость» для анализа устойчивости и режимов движения.
- **Табличные данные:** просмотр численных значений с возможностью экспорта в табличные процессоры.
- **3D-анимация:** визуализация процессов в реальном или модельном масштабе времени.
- **Генерация отчетов:** автоматическое формирование итоговых документов по результатам анализа.

3.1.4 Практические рекомендации по организации работы

Этап работы	Рекомендация
Проектирование схемы	Используйте привязку к сетке (Расположение -> Выравнивать по сетке) для исключения геометрических рассинхронизаций узлов при соединении проводниками.
Массовое редактирование	Для быстрого изменения свойств однотипных компонентов используйте Правка -> Заменить (клавиша F7).
Контроль целостности	Перед расчетом визуально проверяйте связность схемы. Используйте Вид -> Показать все и Масштаб 1:1 для верификации топологии.
Отладка и трассировка	При ошибках запуска обращайтесь к файлу SYSPRINT.txt и просматривайте сгенерированные PPL/PSL файлы (Моделирование -> Показать PPL/PSL файл) для проверки корректности переданных параметров.
Архивирование и передача	Функция Файл -> Экспорт в... позволяет экспортировать схему как паспорт модели для хранения в технической документации или обмена между отделами.
Работа с библиотеками	Используйте строку поиска во вкладке «Компоненты» для быстрой навигации. При работе с многодоменными системами комбинируйте библиотеки, соблюдая правила соединения портов.

3.1.5 Заключение

Препроцессор PRADIS обеспечивает сквозной инженерный цикл: от визуального конструирования физической схемы до автоматической генерации расчетных заданий и передачи данных в постпроцессор. Соблюдение указанной методики гарантирует:

- Минимизацию ошибок на этапе подготовки данных за счет встроенных проверок целостности;
- Стандартизацию рабочих процессов и воспроизводимость результатов;

- Снижение трудозатрат на ручной вывод уравнений благодаря автоматическому формированию системы ДАУ;
- Эффективный анализ динамики сложных технических объектов в широком диапазоне режимов эксплуатации.

Документ составлен на основе официальной документации программного комплекса PRADIS (разделы «Препроцессор», «Руководство пользователя», версия инсталляционного пакета 5.0.071125).

3.2 Руководство по управлению вычислительным ядром Pradis

3.2.1 Введение и назначение

Настоящее руководство предназначено для пользователей программного комплекса PRADIS, выполняющих расчёты динамики технических систем после формирования расчётной модели в препроцессоре. Документ описывает средства управления вычислительным ядром на стадии выполнения анализа, включая настройку точности, контроль процесса интегрирования и диагностику возможных проблем.

Важно

Информация в данном руководстве актуальна для текущей версии PRADIS с графическим интерфейсом (препроцессор PRADIS Qucs). Сведения из документации предыдущих пакетных версий адаптированы с учётом особенностей работы в графической среде.

Вычислительное ядро PRADIS реализует следующие ключевые функции:

- Автоматическое формирование и решение систем дифференциально-алгебраических уравнений (ДАУ) на основе графической схемы модели;
- Численное интегрирование неявными одношаговыми методами (Штермера, Ньюмарка) с автоматическим выбором шага;
- Итерационное решение нелинейных уравнений методом Ньютона с контролем сходимости;
- Оценка и контроль локальной погрешности интегрирования;
- Фильтрация высокочастотных составляющих для оптимизации вычислительных затрат.

3.2.2 Запуск расчёта из препроцессора

Последовательность действий

1. **Подготовка модели:** Убедитесь, что в схеме присутствуют обязательные компоненты:
 - Dynamic — задаёт параметры метода интегрирования и управления расчётом;
 - DISP — определяет параметры для вывода результатов.

2. **Настройка компонента Dynamic:** Откройте свойства компонента двойным щелчком и задайте ключевые параметры (см. раздел 3).
3. **Проверка модели:** Используйте встроенные средства препроцессора для проверки связности схемы и корректности параметров.
4. **Запуск:** Нажмите кнопку «Моделировать» в меню «Моделирование» или используйте клавишу F2.
5. **Мониторинг:** В консоли препроцессора отображается ход расчёта: текущее время, номер шага, сообщения о состоянии процесса.
6. **Завершение:** По окончании расчёта автоматически запускается постпроцессор для визуализации результатов.

Предупреждение

При отсутствии компонента Dynamic или DISP в консоли будет выдано сообщение об ошибке, и расчёт не начнётся.

Оперативное отображение в ходе расчёта

В компоненте Dynamic можно задать список переменных для отображения в реальном времени. Для этого после ключевых параметров через точку с запятой указывается перечень выходных переменных:

```
Скорость груза = (-10, 10), Перемещение = (-2, 2), Сила упругости(2)
```

Формат описания переменной: Имя = (мин, макс) или Имя(номер_компоненты) для многокомпонентных величин.

3.2.3 Ключевые параметры управления ядром

Параметры компонента Dynamic позволяют гибко настраивать поведение вычислительного ядра. Ниже приведена таблица наиболее важных параметров с рекомендациями по их использованию.

Таблица 1: Основные параметры Dynamic

Параметр	Назначение	Диапазон / Значение по умолчанию	Рекомендации
END	Время окончания интегрирования	(обязателен)	Задавайте с запасом относительно ожидаемой длительности процесса
DRLTX	Относительная составляющая допустимой локальной погрешности	(0.001)	0.001-0.01 для предварительных расчётов; 1e-5-1e-4 для окончательных
DABSX	Абсолютная составляющая допустимой локальной погрешности	(0.1)	Задавайте исходя из минимальных амплитуд интересующих переменных
SMAX	Максимальный шаг интегрирования	1e-11 ... (0.01 с)	Увеличивайте для длительных процессов; уменьшайте для быстрых переходных процессов
SMIN	Минимальный шаг интегрирования	1e-11 ... SMAX (1e-11 с)	Не изменяйте без необходимости; увеличение может помочь при проблемах сходимости
ITR	Максимальное число итераций Ньютона на шаге	1 ... (5)	7-10 для сложных нелинейных моделей; 5 достаточно для большинства задач
CONTROL	Нижний предел шага по критерию погрешности; демпфирование ВЧ-колебаний	(0)	Задавайте $\sim 1/10$ минимального периода интересующих колебаний
WEIGHT	Коэффициент демпфирования высокочастотных колебаний	(1.0)	Увеличивайте совместно с уменьшением CONTROL для избирательной фильтрации
OUT	Минимальный интервал сохранения результатов в файл	(0)	Используйте только при очень больших объёмах вывода; не более 1/15 периода быстрых процессов
PRTTIME	Интервал вывода статусных сообщений в консоль (сек)	(30 с)	Уменьшите для детального мониторинга; увеличьте для снижения нагрузки на консоль

Примечание

Параметры DRLTU, DABSU, DRLTI, DABSI управляют точностью решения нелинейных уравнений на каждом шаге. В большинстве случаев значения по умолчанию (0.001 / 0.01) обеспечивают надёжную сходимость.

3.2.4 Управление точностью интегрирования

Алгоритм контроля локальной погрешности

На каждом шаге интегрирования вычислительное ядро оценивает локальную погрешность lp по формуле:

$$lp = |V_{prontos} - V_{korrekciya}|/2$$

где:

- $V_{prontos}$ — значение скорости, полученное явным прогнозом;
- $V_{korrekciya}$ — значение после итерационной коррекции неявным методом.

Допустимая погрешность для каждой степени свободы вычисляется как:

$$[\Delta] = \text{DABSX} + \text{DRLTX} \times |V_{tekuee}|$$

Если $lp \leq [\Delta]$ — шаг принимается, расчёт продолжается. Если $lp > [\Delta]$ — шаг уменьшается и пересчитывается.

Практические рекомендации по настройке точности

- **Предварительные (отладочные) расчёты:** $\text{DRLTX} = 0.01$, $\text{DABSX} = 0.1$ — позволяют быстро получить оценку поведения системы.
- **Окончательные расчёты:** $\text{DRLTX} = 1e-5 \dots 1e-4$, $\text{DABSX} = 1e-6 \dots 1e-4$ — обеспечивают высокую точность.
- **Мультифизические задачи:** Задавайте DABSX исходя из переменных с наименьшими амплитудами (например, давления в гидравлических узлах при наличии механических перемещений).
- **Длительные процессы:** Ужесточайте требования к локальной погрешности, так как ошибка накапливается со временем.

Важно

Значение $\text{DABSX} = 0.1$ по умолчанию соответствует грубому расчёту для процессов с характерными скоростями десятки-сотни единиц. Для точных расчётов всегда явно задавайте оба параметра.

3.2.5 Контроль шага интегрирования

Автоматический выбор шага

Ядро автоматически адаптирует шаг интегрирования Δt на основе:

- Оценки локальной погрешности;
- Сходимости итерационного процесса Ньютона;
- Рекомендаций моделей элементов (например, для учёта разрывов или быстрых изменений).

Рекомендуемый шаг для следующего интервала вычисляется по формуле:

$$\Delta t_{\text{новый}} = c \cdot \Delta t_{\text{текущий}} \cdot \sqrt{(\Delta t)/lp}$$

где $c = 0.8$ — коэффициент запаса.

Явные ограничения: SMAX и SMIN

- **SMAX** — верхняя граница шага. Увеличивайте для длительных квазистатических процессов; уменьшайте для анализа быстрых переходных режимов.
- **SMIN** — нижняя граница шага. Не изменяйте без необходимости. Увеличение может помочь, если расчёт останавливается с ошибкой R 008 (невозможность обеспечить точность на минимальном шаге).

Параметр CONTROL: управление ВЧ-составляющими

Параметр CONTROL выполняет две функции:

1. Задаёт нижний предел уменьшения шага по критерию локальной погрешности (не влияет на ограничения по сходимости или рекомендациям моделей).
2. Обеспечивает принудительное демпфирование высокочастотных колебаний с периодом $< 10 \times \text{CONTROL}$.

Правило настройки CONTROL и WEIGHT:

Для избирательной фильтрации паразитных частот используйте соотношение:

$$\text{CONTROL}^2 \times \text{WEIGHT} = \text{const}$$

Чем меньше CONTROL и больше WEIGHT, тем избирательнее фильтрация (точные НЧ-процессы, сильное гашение ВЧ).

Пример настройки для системы с диапазоном частот:

Если интересуют процессы с периодом ≈ 0.01 с, а более быстрые колебания — паразитные:

```
CONTROL = 0.001    // 1/10 от минимального интересного периода
WEIGHT = 1.0       // стандартное демпфирование
DRLTX = 1e-4       // точность для НЧ-составляющих
DABSX = 1e-5
```

Такая настройка может сократить время расчёта в 10–30 раз при приемлемой точности для интересующих процессов.

3.2.6 Решение нелинейных уравнений на шаге интегрирования**Алгоритм метода Ньютона**

На каждом шаге интегрирования система ДАУ преобразуется в систему нелинейных уравнений $F(U) = 0$, решаемую итерационно:

1. Выбор начального приближения (прогноз по ускорению с предыдущего шага);
2. Вычисление невязки $F(U)$ — суммы усилий от всех моделей элементов;
3. Расчёт приращения $\Delta U = -F/F'$ с использованием якобиана системы;
4. Проверка сходимости по двум критериям:
 - По приращению аргумента: $|\Delta U| < \text{DABSU} + \text{DRLTU} \times |U|$;
 - По невязке: $|F| < \text{DABSI} + \text{DRLTI} \times \sum |F_i|$.
5. При достижении сходимости — переход к контролю погрешности интегрирования; иначе — следующая итерация или уменьшение шага.

Параметры управления сходимостью

Таблица 2: Параметры сходимости

Параметр	Назначение	Рекомендации
ITR	Максимальное число итераций Ньютона	Увеличьте до 7-10 при частых потерях шагов по несходимости. Оптимальный диапазон для большинства задач: 5-7.
SECOND	Требование обязательной второй итерации	Установите SECOND=1, если при ужесточении точности растёт погрешность решения СнЛУ. Гарантирует проверку сходимости даже при быстром достижении формальных критериев.
DABSU / DRLTU	Точность по приращению аргумента	Значения по умолчанию (0.01 / 0.001) подходят для большинства задач. Ужесточайте только при необходимости высокой точности внутренних переменных.
DABSI / DRLTI	Точность по невязке уравнений равновесия	Аналогично: по умолчанию достаточно. Уменьшайте, если требуется строгое соблюдение баланса усилий.

Частые причины плохой сходимости

- Слишком малое значение ITR при больших шагах интегрирования;
- Ошибки в описании структуры модели (некорректные связи, вырожденные элементы);
- Экстремальные значения параметров моделей (например, чрезмерно высокие жёсткости);
- Наличие контактных элементов или сухого трения при больших шагах;
- Ошибки в реализации якобиана пользовательских моделей элементов.

3.2.7 Диагностика и обработка ошибок выполнения

Типовые сообщения и действия

R 006: Несходимость решения системы нелинейных уравнений

Причины: недостаточное число итераций, ошибки в структуре модели, экстремальные параметры.

Действия:

1. Увеличьте ITR до 7-10;
2. Проверьте связность модели и корректность соединений;
3. Уменьшите SMAX для более плавного интегрирования;
4. Используйте параметр DEBUG=11 для вывода отладочной информации о несходимости (см. ниже).

R 008: Минимальный шаг не обеспечивает требуемую точность

Причины: скачкообразные изменения переменных в узлах без инерционных свойств; чрезмерно жёсткие требования к точности.

Действия:

1. Проверьте, нет ли степеней свободы без массы/момента инерции, к которым приложены разрывные воздействия;
2. Используйте параметр CONTROL для ограничения дробления шага в таких участках;
3. Временно увеличьте DABSX/DRLTX для оценки влияния на результат;
4. Рассмотрите введение малых инерционных свойств в «безынерционные» узлы.

Отладочный вывод: параметр DEBUG

Для детальной диагностики включите отладочную информацию в компоненте Dynamic:

```
DEBUG = NNNNLP
```

где:

- NNNN — внутренний номер узла (0 или пусто — все узлы);
- L — уровень вывода: 0 — всегда, 1 — только при проблемах;
- P — тип информации: 1 — решение СнЛУ, 2 — контроль локальной погрешности, 3 — оба типа.

Примеры:

- DEBUG=2 — вывод информации о локальной погрешности для всех узлов на каждом шаге;
- DEBUG=11 — вывод информации о решении СнЛУ только при отсутствии сходимости;
- DEBUG=2413 — детальный вывод по узлу №24 при проблемах с точностью или сходимостью.

Как узнать внутренние номера узлов:

1. Добавьте подраздел #MAP в конец раздела \$FRAGMENT модели;
2. Пересоберите модель — в файл SYSPRINT.TXT будет выведена таблица соответствия пользовательских и внутренних номеров узлов;
3. Используйте внутренние номера при задании DEBUG.

Журнал расчёта

Вся отладочная информация, сообщения об ошибках и статистика расчёта записываются в файл SYSPRINT.TXT. Для просмотра:

- В препроцессоре: меню **Моделирование > Показать журнал**;
- Вручную: файл находится в каталоге проекта .qics\имя_проекта\>.

3.2.8 Управление выводом результатов

Вывод в ходе расчёта (оперативный)

Список переменных, отображаемых в консоли и в реальном времени, задаётся в компоненте Dynamic после ключевых параметров:

```
END=10, DRLTX=0.001; Скорость груза = (-5, 5), Перемещение = (-2, 2), Сила  
→упругости(2) = (-100, 100)
```

Формат: Имя = (мин, макс) или Имя(компонента) = (мин, макс). Если пределы не заданы, используются значения по умолчанию.

Важно

Оперативный вывод не влияет на полноту результатов после расчёта. Все переменные, описанные в компоненте DISP, будут доступны в постпроцессоре независимо от настроек Dynamic.

Сохранение результатов в файл: параметр OUT

По умолчанию значения всех выходных переменных сохраняются на каждом шаге интегрирования в файл *.RSL. Для сокращения объёма файла используйте параметр OUT:

```
OUT = 0.01 // сохранять данные не чаще, чем каждые 0.01 с
```

Рекомендации по использованию OUT:

- Не используйте OUT без необходимости — файлы результатов обычно невелики (десятки КБ);
- Если шаг интегрирования < OUT, параметр игнорируется — данные сохраняются с шагом расчёта;
- Если OUT слишком велик, возможна потеря деталей быстрых процессов. Ориентир: $OUT \approx T_{\min} / 15$, где $T_{\min}$ — период наиболее быстрого интересующего процесса.

Экспорт и постобработка

После завершения расчёта постпроцессор предоставляет возможности:

- **Визуализация:** 2D-графики, фазовые портреты, 3D-анимация;
- **Экспорт данных** в табличные форматы (CSV, TXT) для обработки в сторонних программах;
- **Генерация отчётов** в формате PDF или DOC.

3.2.9 Многоэтапные расчёты и продолжение прерванных сессий

Последовательные вызовы программ интегрирования

В одном задании можно определить несколько вызовов компонента Dynamic с разными параметрами. Каждый последующий вызов продолжает расчёт с момента завершения предыдущего:

```
// Этап 1: разгон, грубая точность
Dynamic_1: END=0.5, DRLTX=0.01, SMAX=0.02;

// Этап 2: рабочий режим, высокая точность
Dynamic_2: END=1.0, DRLTX=1e-4, SMAX=0.005;

// Этап 3: анализ последствий
Dynamic_3: END=2.0, DRLTX=1e-4, CONTROL=1e-4;
```

Преимущества многоэтапной настройки:

- Возможность адаптации точности и шага под разные фазы процесса;
- Устойчивость: сбой на одном этапе не прерывает весь расчёт;
- Гибкость вывода: для каждого этапа можно задать свой список отображаемых переменных.

Продолжение прерванного расчёта: директива \$RESTORE

Если расчёт был прерван (ошибка, останов пользователем, сбой питания), его можно продолжить с последнего сохранённого состояния:

1. В новом задании укажите заголовок \$RESTORE перед разделом \$RUN;
2. Задайте новое значение END, превышающее время последнего успешного шага;
3. Запустите расчёт — ядро загрузит состояние из файлов предыдущей сессии и продолжит интегрирование.

Предупреждение

Запуск расчёта для уже сформированной модели без заголовка \$RESTORE приведёт к потере предыдущих результатов и началу расчёта с нулевого момента времени.

3.2.10 Рекомендации по эффективной настройке

Стратегия настройки для новой модели

1. **Этап отладки:**
 - $DRLTX=0.01$, $DABSX=0.1$, $SMAX=0.01$, $ITR=5$;
 - Цель: быстро проверить работоспособность модели, выявить структурные ошибки.
2. **Этап уточнения:**
 - Ужесточите $DRLTX/DABSX$ до $1e-4$ / $1e-5$;
 - Проанализируйте статистику в `SYSPRINT.TXT`: количество шагов, итераций, потерь;
 - При частых потерях шагов увеличьте ITR до 7-10.
3. **Этап оптимизации:**
 - Если расчёт медленный из-за отслеживания паразитных ВЧ-колебаний — введите `CONTROL` и `WEIGHT`;
 - Для очень длительных процессов настройте `SAVE` для защиты от сбоев;
 - При больших объёмах вывода — аккуратно используйте `OUT`.

Чек-лист перед запуском окончательного расчёта

- Модель содержит компоненты `Dynamiс` и `DISP`;
- Задан обязательный параметр `END`;
- Параметры точности $DRLTX/DABSX$ соответствуют требуемому качеству результата;
- Ограничения шага $SMAX/SMIN$ адекватны длительности и динамике процесса;

- При наличии быстрых/разрывных воздействий задан параметр CONTROL;
- Для сложных нелинейных моделей ITR ? 7;
- Для длительных расчётов задан SAVE;
- Список оперативного вывода не перегружен (не более 5–7 переменных).

Типовые значения параметров для распространённых задач

Таблица 3: Типовые параметры

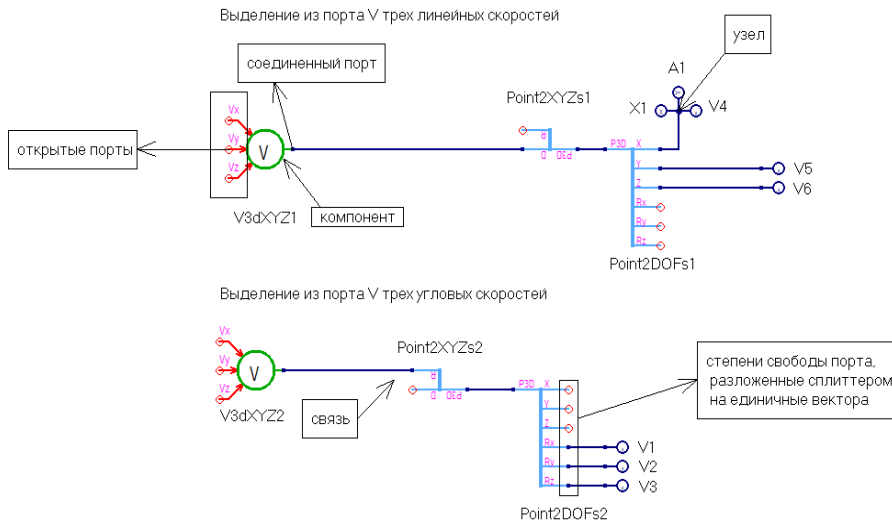
Тип задачи	DRLTX	DABSX	SMAX	CONTROL	ITR
Предварительный анализ, отладка	0.01	0.1	0.01	0	5
Механические колебания (НЧ)	1e-4	1e-5	0.005	T _{мин} /10	7
Ударные/контактные задачи	1e-5	1e-6	0.001	1e-6	10
Длительные процессы с ВЧ-фильтрацией	1e-4	1e-5	0.02	0.001	7
Мультифизические (механика + гидравлика)	1e-5	1e-7	0.005	по НЧ-процессам	7–10

3.3 Степени свободы, сплиттеры и методы построения многодоменных схем

3.3.1 Введение и назначение

Препроцессор PRADIS реализует объектно-ориентированный подход к моделированию динамики технических систем. Математическая основа любой схемы базируется на узлах (Nodes) и их степенях свободы (DOF), которые определяют размерность и физическую природу расчётного пространства. Для соединения подсистем с различной размерностью, преобразования координат или выделения отдельных физических переменных применяются специализированные интерфейсные компоненты – сплиттеры (splitters). Настоящее руководство объединяет теорию базовых узлов, описание сплиттеров и практическую методику их применения. Документ предназначен для инженеров-расчётчиков, создающих модели сложной структуры, в том числе многодоменные (механика + гидравлика + автоматика + акустика) и гибридные (2D/3D) схемы.

Основные понятия, используемые в PRADIS:



Степень свободы в PRADIS соответствует одной потенциальной переменной узла, для которой в уравнениях модели формируется сопряженная потоковая переменная: сила, момент, расход, ток или иной обобщенный поток в зависимости от физического домена. **Порт** – конечная совокупность степеней свободы. Каждый компонент имеет минимум один порт. Объект на схеме, соединяющий порты разных компонентов, называется **связью** (проводником в терминах препроцессора PRADIS), а соединение связей на схеме называется **узлом**. При этом порт, исходящая из него связь и узел всегда имеют одинаковое количество степеней свободы. **Сплиттер** – объект, который позволяет соединять проводники с разным числом степеней свободы.

Обобщение основных понятий:

Объект	Назначение
Порт	Интерфейс компонента
Связь	Соединение портов
Узел	Топологическое объединение связей
DOF	Элементарная степень свободы
Сплиттер	Преобразователь структуры DOF

3.3.2 Узлы и степени свободы в PRADIS

Концепция DOF

Каждый компонент препроцессора соединяется с другими через порты, которые привязаны к математическим степеням свободы. Тип порта определяет:

- Количество независимых координат (поступательных и вращательных);
- Физический домен (механика, термодинамика, акустика, логика);

Таким образом, для каждого типа узла состав потенциальных и потоковых переменных, размерность расчётного пространства и структура вкладов элемента в расчётную систему PRADIS определяются типом портов для соответствующих связей. Узлы могут быть представлены одной или несколькими элементарными степенями свободы, которые описываются базовым типом DOF, что обеспечивает строгую совместимость уравнений при

3.3. Степени свободы, сплиттеры и методы построения многодоменных схем

автоматической генерации системы ОДУ.

Классификация и описание узлов

Узел	Назначение	Число степеней свободы	Состав элементов	Физический смысл
DOF	Узел с одномерной степенью свободы	1	x	Простая степень свободы, характеризующаяся перемещением по одному направлению, электрическим потенциалом, давлением и т.п.
XY	Узел двухмерного поступательного движения	2	x, y	Плоское поступательное перемещение без вращения.
XYZ	Узел поступательного 3D движения	3	x, y, z	Пространственное поступательное перемещение. Распределение фаз в трехфазной электрике.
Point2d	Узел планарного движения точки	3	x, y, R	Плоское движение с вращением (X, Y + R). Стандарт для 2D-механизмов.
Point	Узел 3D движения точки	6	x, y, z, gx, gy, gz	Полное пространственное движение твердого тела/точки.
BUS	Узел шины	0	(отсутствуют)	Логическая коммутация. Группировка портов, передача управляющих сигналов без физической инерции.
Thermal	Тепловая гидравлика	4	p, h, T	Работа с давлением, энтальпией, плотностью и температурой среды.
Thermal2Mi	Смесевая гидравлика	5	p, h, T, x	Работа с давлением, энтальпией, плотностью и температурой среды, а также с массовой долей первой компоненты.

Дополнительная информация по узлам и степеням свободы содержится по ссылке (актуальная ссылка после загрузки в git)

Рекомендации по выбору узла

- DOF – оптимальный выбор для задач с одной обобщенной координатой (пружины, демпферы, поршни, электрические ветви).
- XY – плоские системы без учета поворота звеньев (ползуны, направляющие, деформации плоских конструкций).
- Point2d – планарные механизмы с вращением (кривошип, рычаги, плоские манипуляторы).
- Point – полное 3D-моделирование с учетом поступательных и вращательных степеней свободы (балки, стержни, пространственные пружины). Для пространственных механических узлов Point угловые степени свободы требуют отдельного внимания, поскольку в PRADIS они могут описываться через специальные кинематические параметры. Более подробно об угловых степенях свободы при моделировании 3d движения можно прочитать здесь (актуальная ссылка после загрузки в git)
- BUS – логический/структурный интерфейс без собственных физических степеней свободы

- ThermalFluid2 – работа со средами, состоящими из одной фазы с контролем давления, плотности, температуры, энтальпии (дроссели, гидроаккумуляторы, гидроцилиндры, насосы)
- Thermal2MixFluid2 – работа со средами, состоящими из двух фаз с контролем давления, плотности, температуры, энтальпии и массовой доли первой компоненты (дроссели, дегазатор/аэрактор, насосы)

3.3.3 Сплиттеры: инструменты согласования и декомпозиции (splitters)

Назначение и принцип работы

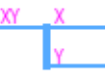
Сплиттеры – это интерфейсные компоненты, выполняющие роль математических адаптеров. Большинство сплиттеров не вводят собственных инерционных свойств; некоторые адаптеры размерности могут вводить параметризованную жесткую или упругую связь между подсистемами. Сплиттер не должен использоваться как физическая модель деформации, если его паспорт описывает только преобразование координат или декомпозицию DOF.

Назначение сплиттеров:

- Декомпозируют составные узлы на элементарные степени свободы (DOF);
- Преобразуют координаты между глобальной и локальными системами отсчёта (ЛСК);
- Извлекают отдельные термодинамические, акустические или потоковые переменные;

1. Классификация сплиттеров

Сплиттер	Функция	Свойства
Кинематические разветвители (Декомпозиция механических узлов на элементарные порты DOF или субузлы)		

 Point2DOFs	{P3D(Point)}{X, Y, Z, Rx, Ry, Rz}	
 XYZ_X_Y	{XY}{X, Y}	
 Point2d2XY_R	{P2D}{XY, R}	
 Point2XYZs	{P3D}{D, R}	
 XYZZ_X_Y_Z	{XYZ}{X, Y, Z}	
 Point2d2DOFs	{P2D}{X, Y, R}	
Разделение плоского движения на компоненты.		

Адаптеры размерности (Жёсткая/упругая связь плоских (2D) и пространственных (3D) подсистем)

 Point2d2Point	{P2D}{P3D}	Kx, Ky, Kt (жёсткости, по умолч.1e6); PointO, PointX, PointXoY (ориентация 2D-плоскости в 3D)
--	------------	---

Преобразователи координат (ЛСК) (Трансформация координат между ГСК и ЛСК через матрицу поворота/смещения)

 obj_P1DTP2D	 obj_P2DTP2D	 obj_P3D2P3D	LCS0 - матрица поворота/смещения новой ЛСК относительно ГСК
--	--	--	---

Теплогидравлические (Выделение давления p, температуры T, энтальпии h, плотности го, концентрации x ** *(В отличие от датчиков, обладающих сопротивлением, не влияет на параметры системы.))

 ThermalFluid2PT	{Fluid (ThermalFluid)}{p, T}	Базовый контроль давления и
--	------------------------------	-----------------------------

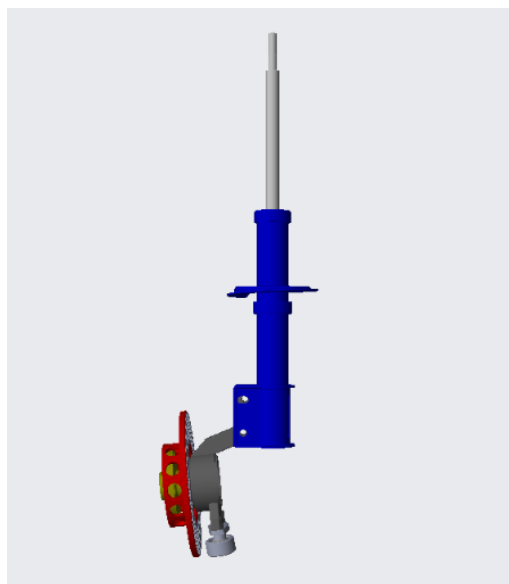
Подключение линейного демпфера (DOF) к пространственной массе (Point)

Пример с упрощенной амортизаторной стойкой

Описание модели

Модель элемента с начальным перемещением предназначена для анализа поведения пространственного инерционного элемента в случае начального перемещения по оси z , после чего формируется движение объекта с учетом его инерционных свойств. Линейный демпфер работает с одномерным узлом DOF. Пространственная масса работает с узлом Point. Для соединения этих элементов необходимо выделить из Point соответствующую степень свободы. Эту задачу выполняет Point2DOFs. Система отслеживает положение объекта в заданный отрезок времени.

На практике такую систему можно представить в виде работы упрощенной амортизаторной стойки, расчет которой позволяет выявить амплитуду перемещения измеряемого элемента за заданный промежуток времени, а также получить диаграмму результатов расчетов.



Параметры модели

S_0 — начальное перемещение, м; $S_0 = 0,07$.

mass — масса объекта, кг; mass = 60.

moments - главные моменты инерции относительно локальных осей; moments = 1.0, 1.0, 1.0.

K - коэффициент упругости, Н/м; K=2500.

SV1MUT- вязкость с задаваемой таблично зависимостью усилия от скорости, м/с, Н*с/м (попарно); -1, -500, 0, 0, 1, 150.

Элементы модели

Начальное перемещение X0 — задаёт перемещение по оси Z.

Пространственный инерционный элемент MJ3D1— механический объект, положение которого отслеживается.

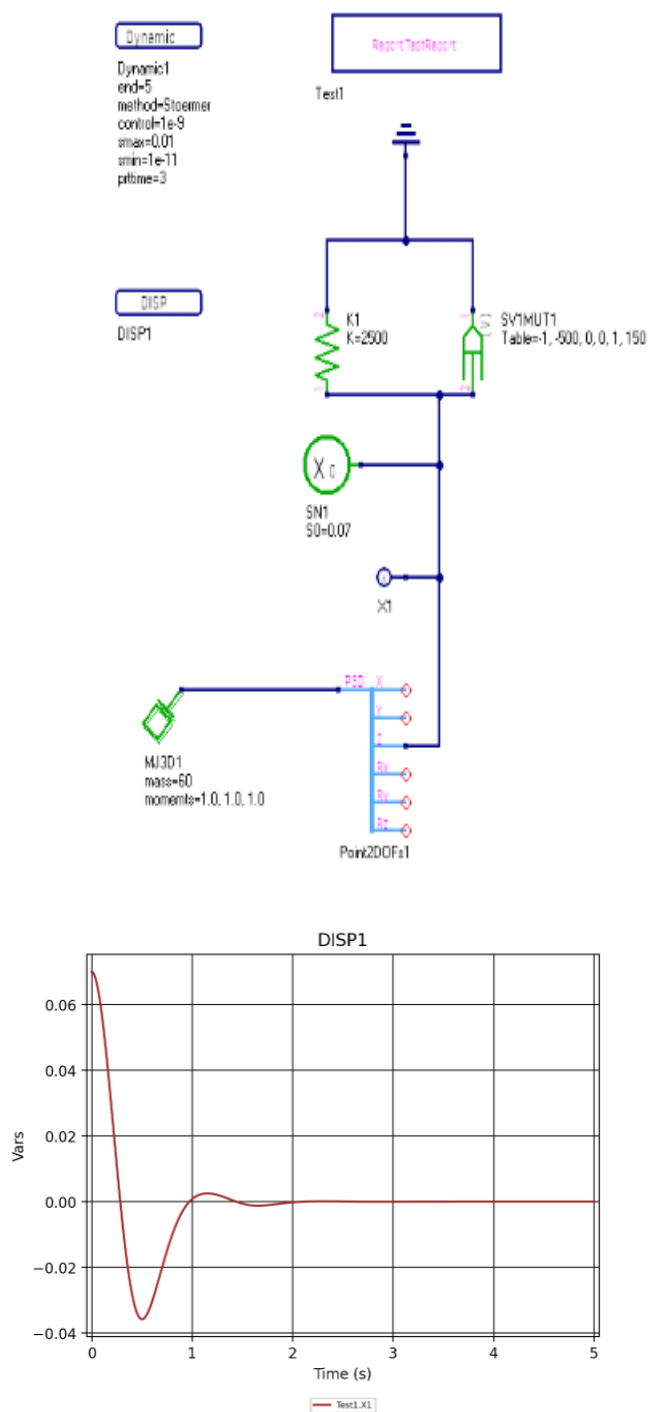
Point2DOFs - разветвитель узла Point по степеням свободы x , y , z , R_x , R_y , R_z .

Датчик перемещения X1— измеряет перемещение объекта во времени.

SV1MUT1 - одномерная вязкостная связь с задаваемой таблично зависимостью усилия от скорости.

K1 - элемент упругости.

Схема модели в PRADIS и полученные графики.

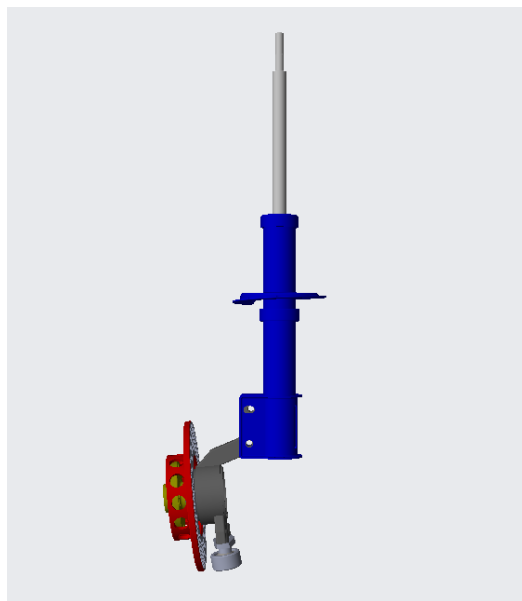


На графике мы наблюдаем переходной процесс изменения перемещения заданной массы от 0,07 м за время 5 с.

Пример с амортизаторной стойкой с учетом бокового наклона профиля дороги

Описание модели

В модели с пространственным инерционным элементом и шестью характеристиками в виде упругости и вязкости применяется постоянный источник силы, задается начальное перемещение. На практике такую систему можно представить в виде работы амортизаторной стойки, в случае, когда профиль дороги имеет боковой наклон и автомобиль наезжает на препятствие определенной высоты. Расчет позволяет выявить скорость и амплитуду перемещения измеряемого элемента, а также получить диаграмму результатов расчетов.



Параметры модели

S_0 — начальное перемещение, м; $S_0 = 0,07$.

mass — масса объекта, кг; mass = 60.

moments - главные моменты инерции относительно локальных осей; moments = 1.0, 1.0, 1.0.

K - коэффициент упругости, Н/м; K=2500.

SV1MUT- вязкость с задаваемой таблично зависимостью усилия от скорости, м/с, Н*с/м (попарно); -1, -500, 0, 0, 1, 150.

LongitudinalForce - продольная сила, Н; FC=2000.

Mu1 - вязкость, Н*с/м; Mu = 800.

LateralForce - поперечная сила, Н; FC=8000.

Mu2 - вязкость, Н*с/м; Mu = 800.

Элементы модели

Начальное перемещение X_0 — задаёт перемещение по оси Z.

Пространственный инерционный элемент MJ3D1 — механический объект, положение которого отслеживается.

Point2DOFs - разветвитель узла Point по степеням свободы x , y , z , R_x , R_y , R_z .

Датчики перемещения X1, X2, X3 — измеряют перемещение объекта во времени.

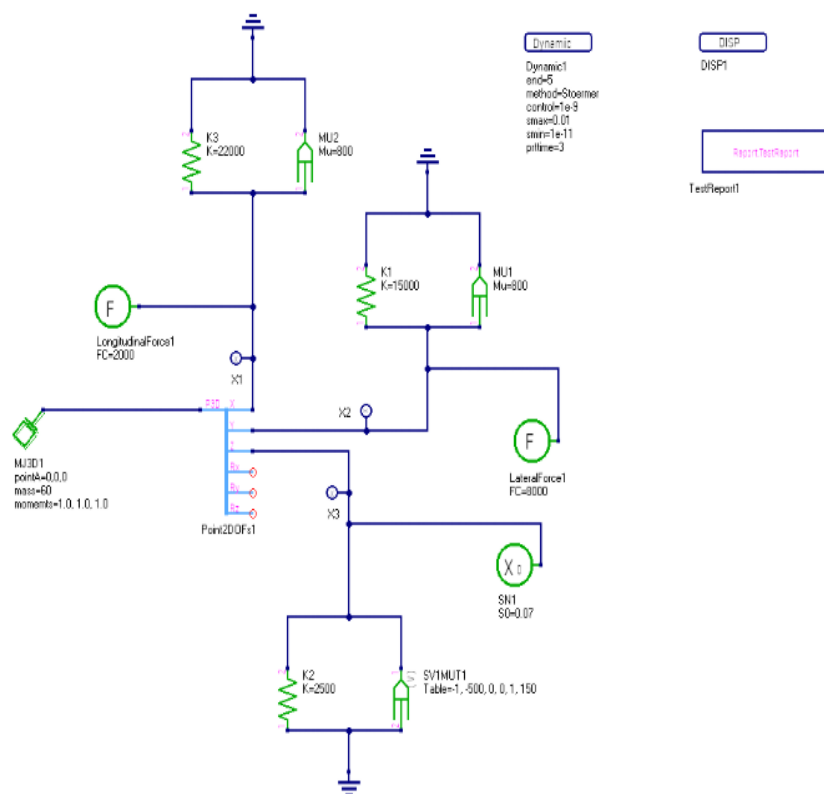
Источники постоянной силы F - определяют величину прикладываемой силы.

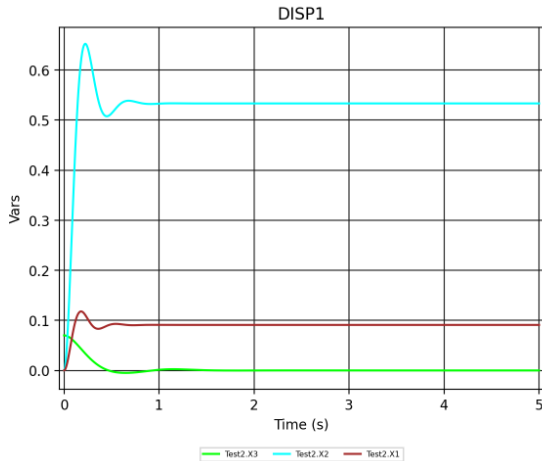
SV1MUT1 - одномерная вязкостная связь с задаваемой таблично зависимостью усилия от скорости.

K1, K2, K3 - элементы упругости.

Mu1, Mu2 - компоненты вязкости.

Схема модели в PRADIS и полученные графики.





На графике мы видим, как при приложении силы к поворотному кулаку происходит перемещение колеса в поперечном и продольном направлении за счет деформации стойки. В вертикальном направлении через пять секунд колесо возвращается в исходное положение.

Использование сплиттеров P2DTP2D / P3D2P3D

Моделирование наклонного рельса или направляющей (P2DTP2D)

Описание модели

Тело движется вверх по наклонной поверхности. На него действуют сила тяжести, направленная вертикально вниз, и сила тяги, направленная под определенным углом. Мы применяем сплиттер P2DTP2D, позволяющий сориентировать ось Ox локальной системы координат в направлении вектора силы тяги. Система отслеживает положение объекта в заданный отрезок времени. На практике это можно представить как упрощенную модель взаимодействия тела и наклонной поверхности.

Параметры модели

angle — начальный угол наклона направляющих к оси Ox , градусы; angle = 45.

mu — Коэффициент трения, -; mu = 0,1.

FC - Значение силы/момента постоянной величины, Н; FC=1000.

M - Масса тела, кг; M=100

g - Номер оси, вдоль которой действует сила тяжести (1-3), -; g=2

служебные коэффициенты

LCS0 = [1,0,0,0,1,0,0,0,1] - положение старой ЛСК (ЛСК - локальная система координат), где

Ox Oy сдвиг системы координат

(1, 0) - направляющие косинусы Ox , (0, 1) - направляющие косинусы Oy ,

(0, 0) - сдвиг системы координат.

(0, 0, 1) - служебные коэффициенты для возможности объединения поворота и сдвига в одной матрице. Они всегда равны (0, 0, 1).

LCS0 = [b, b, 0, bm, b, 0, 0, 0, 1] соответствует новой ЛСК, где аналогично:

(b, b) - направляющие косинусы O_x, (bm, b) - направляющие косинусы O_y,

(0, 0) - переменные для перемещения новой ЛСК;

(0, 0, 1) - служебные коэффициенты для возможности объединения поворота и сдвига в одной матрице. Они всегда равны (0, 0, 1).

$b = 1 / \sqrt{2}$, $bm = -b$.

Проверка новой системы координат на ортогональность:

**Условие перпендикулярности векторов
в координатной форме :**

$$\vec{a} \perp \vec{b} \Leftrightarrow x_1 x_2 + y_1 y_2 + z_1 z_2 = 0$$

Мы должны проверить, что O_x и O_y перпендикулярны.

O_x и O_y: $(1 / \sqrt{2}) * (1 / \sqrt{2}) + (-1 / \sqrt{2}) * (1 / \sqrt{2}) = 0$.

Элементы модели

MJ2EG1 - 2D инерционный элемент со смещенным центром с учетом силы тяжести.

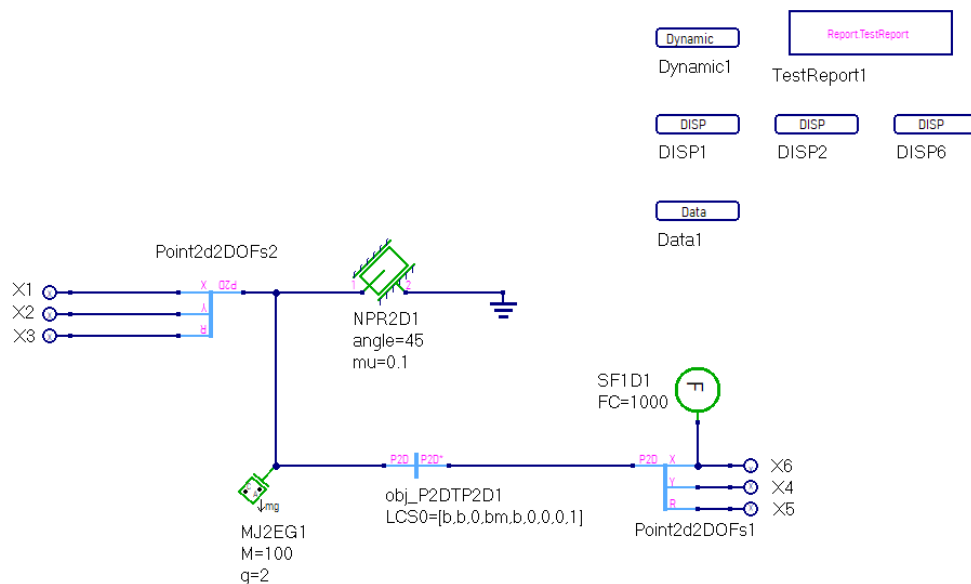
NPR2D1 - направляющие скольжения, перемещающиеся в плоскости, идеально упругие, с трением и зазором.

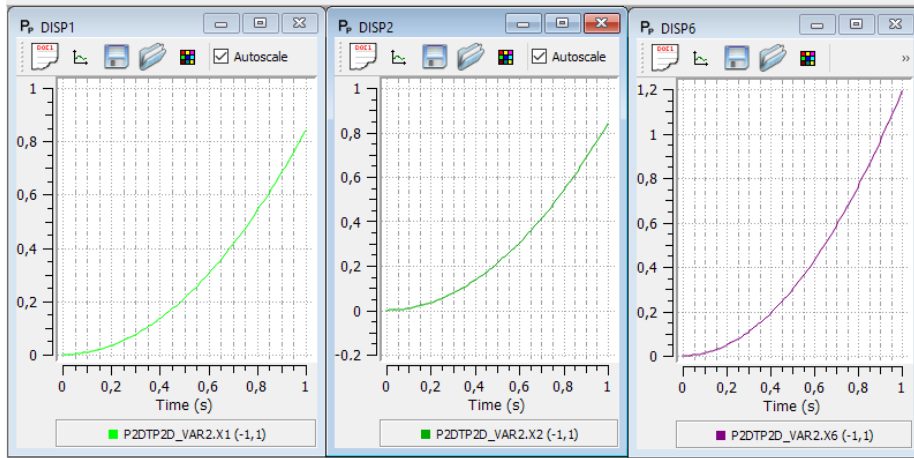
F - источник силы/момента постоянной величины.

Point2d2DOFs2 - разветвители узла Point2d по осям x, y, r.

Obj_P2DTP2D - преобразование исходной 2D ЛСК в новую 2D ЛСК

Схема модели в PRADIS и полученные графики.





По графикам мы можем определить, что величина перемещения тела с датчика х6, взятая в момент времени 1 с, равна корню квадратному из суммы квадратов значений перемещения тела по осям х и у, взятых с датчиков х1 и х2 в момент времени 1 с, что совпадает с расчетной величиной перемещения.

Расчет проекций скоростей тела в новой системе координат (P3D2P3D)

Описание модели

В исходной 3D ЛСК тело массой M движется со скоростью V , проекции которой на оси Ox , Oy и Oz заданы изначально. Затем исходная ЛСК преобразуется в новую путём ее поворота на 90 градусов против часовой стрелки вокруг оси Z при помощи сплиттера P3D2P3D. В новой системе координат мы получаем новые проекции скорости. (см. Приложение 2)

Параметры модели

obj_Const_x: $C=4$, obj_Const_y: $C=2$, obj_Const_z: $C=5$ – значения констант для источников постоянного сигнала. То есть мы задаем значения скорости по x , y и z соответственно равными 4, 2 и 5.

mass — масса объекта, кг; $mass1 = 1$.

Служебные коэффициенты

$LCS0 = [1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]$ – положение старой ЛСК, где

$Ox Oy Oz$ сдвиг системы координат

(1, 0, 0) - направляющие косинусы Ox ,

(0, 1, 0) - направляющие косинусы Oy ,

(0, 0, 1) - направляющие косинусы Oz ,

(0, 0, 0) – сдвиг системы координат.

(0, 0, 0, 1) – Служебные коэффициенты для возможности объединения поворота и сдвига в одной матрице. Они всегда равны (0, 0, 0, 1).

$LCS0 = [0, 1, 0, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]$ - положение новой СК, где аналогично:

(0, 1, 0) - направляющие косинусы Ox ,

(-1, 0, 0) - направляющие косинусы Oy ,

(0, 0, 1) - направляющие косинусы Oz ,

(0, 0, 0) – сдвиг системы координат.

(0, 0, 0, 1) - Служебные коэффициенты для возможности объединения поворота и сдвига в одной матрице. Они всегда равны (0, 0, 0, 1).

Проверка новой системы координат на ортогональность:

Условие перпендикулярности векторов
в координатной форме :

$$\vec{a} \perp \vec{b} \Leftrightarrow x_1x_2 + y_1y_2 + z_1z_2 = 0$$

- условие перпендикулярности векторов в координатной форме. Мы должны проверить, что Oх и Oу, Oу и Oz, Oх и Oz попарно перпендикулярны.

Oх и Oу: $0 \cdot (-1) + 1 \cdot 0 + 0 \cdot 0 = 0$; Oу и Oz: $(-1) \cdot 0 + 0 \cdot 0 + 0 \cdot 1 = 0$; Oх и Oz: $0 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 = 0$.

При заданной ориентации новой ЛСК проекции скорости должны удовлетворять соотношениям: $V_{x_out} = V_{y_in}$, $V_{y_out} = -V_{x_in}$, $V_{z_out} = V_{z_in}$ — при принятой в примере ориентации осей.



Элементы модели

obj_Const1, obj_Const2, obj_Const3 – источники постоянного сигнала

V – источник 3D скорости

Ball1 – инерционный однородный шар, работающий с геометрическим объектом привязки.

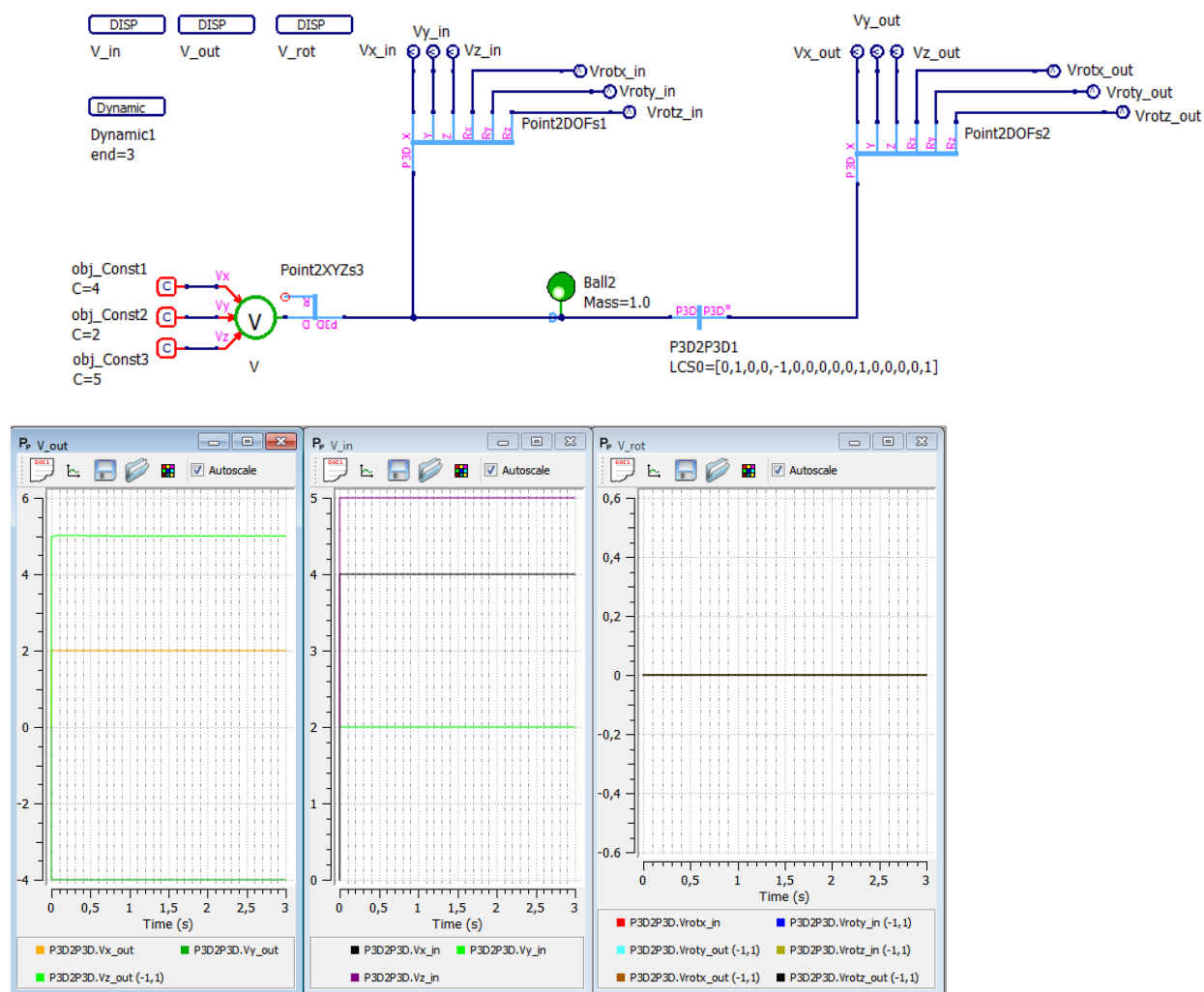
Point2XYZs3 – разветвитель узла x, y, z по степеням свободы x, y, z.

Point2DOFs1, Point2DOFs2 – разветвители узла Point по степеням свободы x, y, z, Rx, Ry, Rz.

P3D2P3D1 – пересчет движения точки из глобальной в локальную систему координат.

V_{x_in} , V_{y_in} , V_{z_in} , V_{x_out} , V_{y_out} , V_{z_out} , V_{rotx_in} , V_{roty_in} , V_{rotz_in} , V_{rotx_out} , V_{roty_out} , V_{rotz_out} – индикаторы масштабированного значения скорости в узле.

Схема модели в PRADIS и полученные графики.



Результат моделирования:

По графикам мы можем определить, что значение скорости по оси Oz в новой системе координат не поменялось, значение скорости по оси Ox стало равно значению скорости по оси Oy, а значение скорости по оси Oy стало равно отрицательному значению скорости по оси Ox, что соответствует повороту старой ЛСК на 90 градусов против часовой стрелки вокруг оси Z.

Связь плоского механизма (Point2d) с 3D-основанием (Point)

Описание модели

Трехмерное тело с массой равной $mass$ движется с постоянной скоростью V , вектор которой задан в плоскости 2D-ЛСК и сонаправлен с осью Z. При помощи параметров сплиттера Point2d2Point мы можем расположить 2D-ЛСК в 3D-СК необходимым нам образом.

Параметры модели

PointO - Начало координат 2D-узла в 3D-системе координат.

PointX - Точка на оси X 2D-узла, определяющая направление оси Ox.

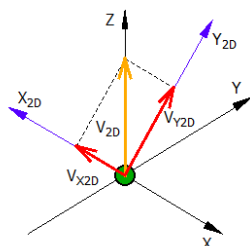
PointXoY - Точка на плоскости XoY 2D-узла, задающая плоскость XOY.

Kx - Жесткость в направлении X

Ky - Жесткость в направлении Y

Kt - Жесткость на кручение

Kx= Ky= Kt=1e6.



В данном случае мы изменяем значения по умолчанию указанных параметров сплиттера на следующие:

PointO [0,0,0] -> [0,0,0]

PointX [1,0,0] -> [0,-1,1]

PointXoY [0,1,0] -> [0,1,1]

Тогда:

$VZ_3D = V2D = (VX2D^2 + VY2D^2)^{0.5}$

Mass — масса объекта, кг; Mass = 2.

obj_Const1: C=2, obj_Const2: C=2- значения констант для источников постоянного сигнала. То есть мы задаем значения скорости по x и y соответственно равными 2 и 2.

Элементы модели

obj_Const1, obj_Const2, obj_Const3 - источники постоянного сигнала.

V2d - источник 3D скорости.

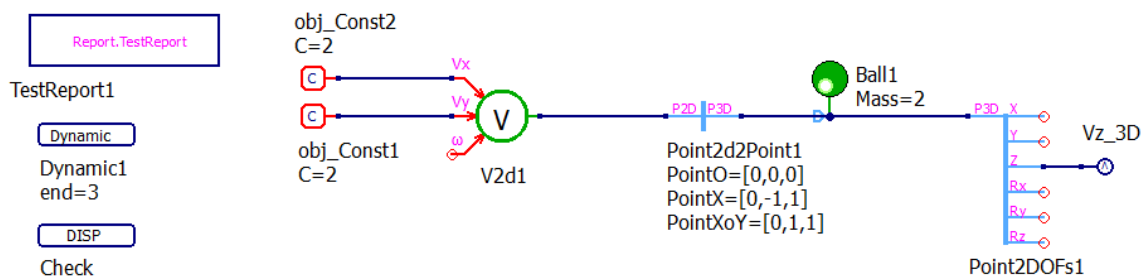
Ball1 - инерционный однородный шар, работающий с геометрическим объектом привязки.

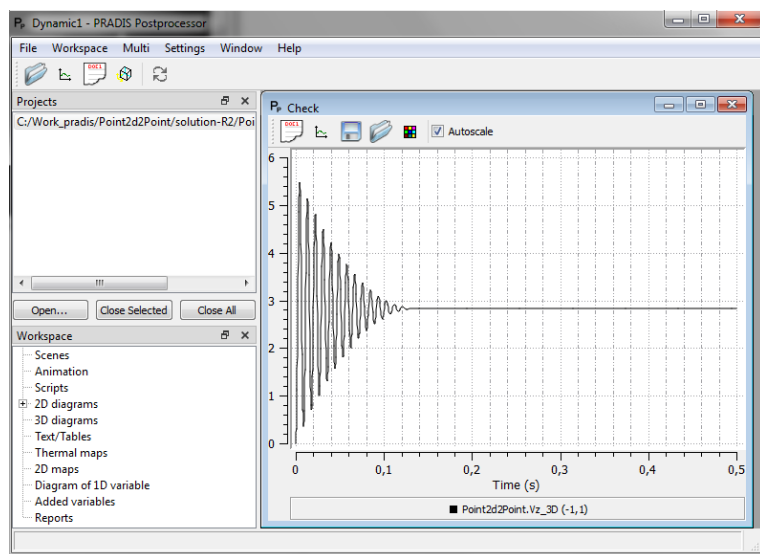
VZ_3D - индикатор масштабированного значения скорости в узле.

Point2d2Point - связь между узлами «Point2d» (2D-точки) и «Point» (3D-точки).

Point2DOFs - разветвитель узла «Point» по степеням свободы x, y, z, Rx, Ry, Rz.

Схема модели в PRADIS и полученные графики.





Результат моделирования:

Значения на графике соответствуют формуле.

Отбор давления для ПИД-регулятора гидросистемы

Описание модели

Использование сплиттера ThermalFluid2PT для модели с источником переменного давления и стоком. Имеется труба с тремя портами, к двум портам типа Thermalfluid подсоединены источники постоянного давления, а тепловой порт в данной задаче не используется. За счет разницы давлений происходит движение жидкости в трубе. С помощью сплиттера ThermalFluid2PT, подключенного к точке A, мы можем выделить значение объемного расхода по линии P и значение потока энтальпии по линии T для дальнейшего анализа этих значений.

Параметры модели

P — давление, кПа; $P_1=200$, $P_2=180$.

T — Температура, C; $T_1=T_2=30$.

obj_Const1: $C=30$ — значение температуры для управляемого источника давления.

obj_TrapeziumSource1:

VL - Пассивный уровень; $VL=180$

VH - Активный уровень; $VH=200$

D - Начальная задержка; $D=0.1$

FT - Продолжительность переднего фронта импульса; $FT=0,5$

Rt - Внутреннее сопротивление источника температуры; $R=1e-6$.

Rp - Внутреннее сопротивление источника давления; $R=1e-6$.

Модуль упругости E_g , абсолютная вязкость Nu и плотность Ro жидкости импортируются из файлов типа *.f2a через элементы ImportData.

Элементы модели

SPGT1—источник постоянного давления.

USPGT1 - управляемый источник давления.

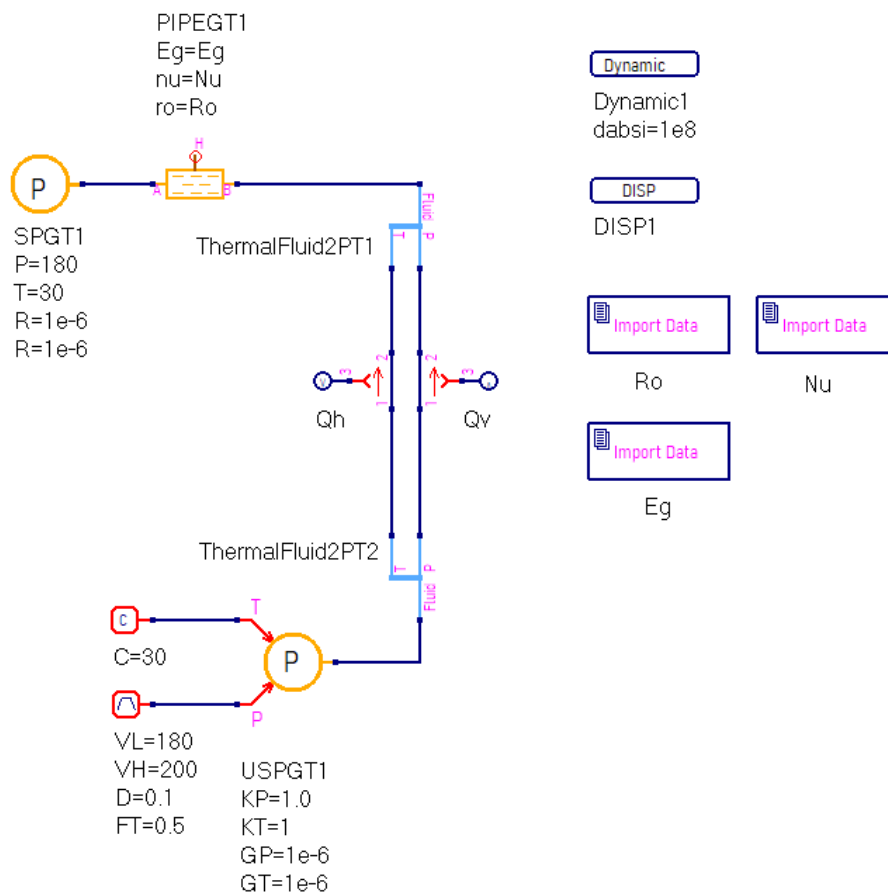
ThermalFluid2PT1, ThermalFluid2PT2 - разветвители узла по тепловой гидравлике для давления и температуры.

PIPEGT1 - труба с теплопереносом.

obj_Const1- источник постоянного сигнала.

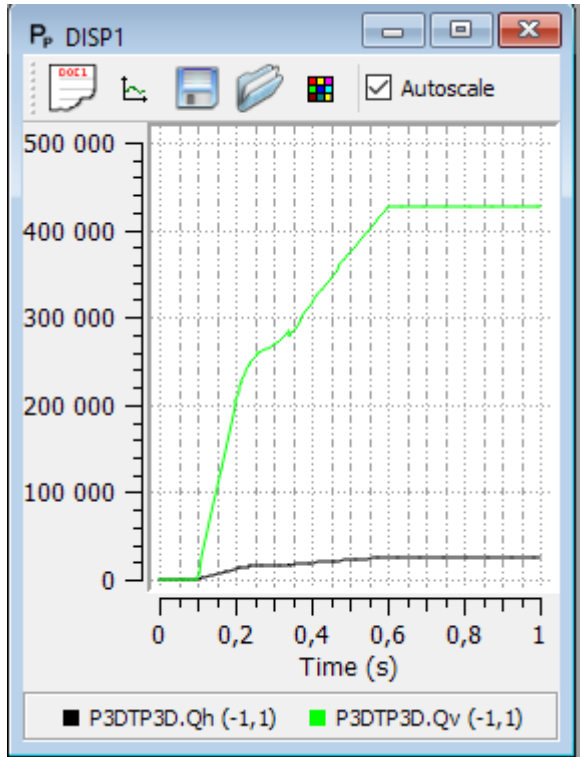
obj_TrapeziumSource1 - источник сигнала трапециевидной формы.

Qh, Qv - преобразователи разницы сил в сигнал.



модели в PRADIS и полученные графики.

Схема



На графиках мы наблюдаем процесс изменения потока энтальпии и объемного расхода жидкости в трубе за время, равное 1 с, при увеличении перепада давления.

3.3.4 ПРИЛОЖЕНИЕ 1 - Угловые степени свободы в пространственных элементах

Кинематические параметры

Согласно теореме Эйлера, твёрдое тело из одного углового положения в другое можно перевести одним поворотом вокруг некоторой оси. Вводятся четыре кинематических параметра:

$$x_1 = e_1 \sin \frac{\Phi}{2}, \quad x_2 = e_2 \sin \frac{\Phi}{2}, \quad x_3 = e_3 \sin \frac{\Phi}{2}, \quad x_4 = \cos \frac{\Phi}{2} \quad (1)$$

Эти параметры связаны уравнением:

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 = 1 \quad (2)$$

В отличие от любой совокупности трёх кинематических параметров (например, углов Эйлера), указанные четыре параметра не вырождаются при любом положении твёрдого тела.

Угловые степени свободы в PRADIS

$$q_1 = x_1 L_q, \quad q_2 = x_2 L_q, \quad q_3 = x_3 L_q, \quad q_4 = x_4 L_q \quad (3)$$

где

$$L_q = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (4)$$

Первые три степени свободы являются внешними для моделей элементов, четвертая — внутренняя, скрытая от пользователя. Начальное значение потенциальной переменной, соответствующей внутренней степени свободы, устанавливается в моделях элементов равным 1.

Потоковые переменные

Потоковыми переменными для первых трёх степеней свободы являются моменты по глобальным осям X, Y, Z. Четвёртая (внутренняя) потоковая переменная сдерживает изменение во времени величины L_q :

$$i_4 = M_u \frac{dL_q}{dT} \quad (5)$$

где M_u — коэффициент пропорциональности, одинаковый для всех степеней свободы такого рода и зависящий от заданной точности решения и разрядной сетки используемого процессора.

Рекомендации для пользователя

Для пользователя корректны практически все приёмы, характерные для поступательно-го движения:

- Можно запрещать движение по выбранным угловым степеням свободы (базируя соответствующие узлы), что равносильно сокращению размерности вектора, направленного по оси конечного вращения.
- Связь по вращению между точками стыкуемых элементов можно осуществлять не по всем трём степеням свободы, а только по выбранным.

Внимание: в отличие от плоского вращения, первая и вторая производные от потенциальных переменных (3) не являются угловой скоростью и угловым ускорением соответственно. Начальные условия, задаваемые моделью VN, не определяют, в общем случае, начальную угловую скорость. ПРВП типов V и A выводят текущие значения первой и второй производной от потенциальной переменной. Значения угловых скоростей и ускорений доступны из рабочего вектора некоторых моделей элементов, в частности J3O.

3.3.5 ПРИЛОЖЕНИЕ 2

Для сплиттера P2DTP2D LCS0 = [a1, a2, a3, a4, a5, a6, a7, a8, a9] соответствует матрице:

$$\begin{pmatrix} a1, & a4, & a7, \\ a2, & a5, & a8, \\ a3, & a6, & a9 \end{pmatrix},$$

в которой:

$$\begin{pmatrix} a1, & a4 \\ a2, & a5 \end{pmatrix},$$

—;

(a1, a2) - направляющие косинусы Oх,

(a4, a5) - направляющие косинусы Оу,

(a7, a8) – сдвиг системы координат:

(a3, a6, a9) - служебные коэффициенты для возможности объединения поворота и сдвига в одной матрице. Они всегда равны (0, 0, 1).

Для сплиттера P3DTP3D LCS0 = [a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, a16] соответствует матрице:

(a1 a5 a9 a13

a2 a6 a10 a14

a3 a7 a11 a15

a4 a8 a12 a16), в которой:

(a1 a5 a9

a2 a6 a10

a3 a7 a11) - Ядро матрицы, отвечающее за поворот;

(a1, a2, a3) - направляющие косинусы Oх,

(a5, a6, a7) - направляющие косинусы Oу,

(a9, a10, a11) - направляющие косинусы Oz,

(a13, a14, a15) - сдвиг системы координат:

(a4, a8, a12, a16) - служебные коэффициенты для возможности объединения поворота и сдвига в одной матрице. Они всегда равны (0, 0, 0, 1).

3.4 Руководство по включению в библиотеки комплекса PRADIS моделей элементов и объектов

3.4.1 Назначение и область применения

Руководство описывает порядок подготовки и включения в библиотеки комплекса PRADIS моделей элементов и объектов. Под включением понимается не только копирование файлов в каталоги DINAMA, но и согласование программной реализации, XML-паспортов, регистрационных файлов, справки, тестов и проверок качества.

Документ рассчитан на разработчиков пользовательских моделей, администраторов поставки, сопровождающих библиотек и QA-специалистов. Для первого подключения достаточно пройти разделы 2-10. Разделы 11-14 используются при разработке моделей на конкретных языках и при разборе ошибок.

3.4.2 Как пользоваться руководством

1. Определите, что именно включается: библиотека расчетных моделей, отдельная модель, Python-объект или комплект, содержащий несколько сущностей.
2. Сверьте комплект поставки по разделу 5: без DLL, XML-паспортов, служебных описаний и теста подключение считается неполным.
3. Разместите файлы по структуре раздела 6 и выполните ручную или автоматизированную регистрацию по разделам 8-9.
4. Проверьте plugin_repository.xml, потому что именно там связываются пользовательский паспорт, путь к библиотеке и имя вызываемой процедуры.
5. Выполните минимальный расчетный тест и зафиксируйте результат в TestReport или принятом внутреннем формате.

3.4.3 Роли и ответственность

Роль	Основные действия	Контроль результата
Разработчик	Формулирует модель, пишет код, готовит DLL/скрипт, XML-паспорта, справку и тест.	Имена модели, DLL, procedure, портов, параметров и рабочих переменных согласованы.
Администратор / сопровождающий	Размещает файлы в DINAMA, обновляет sysarm.xml и plugin_repository.xml, запускает armdoc/arm или утвержденный скрипт.	Библиотека видна, путь к DLL корректен, procedure указывает на экспортируемую функцию.
QA	Проверяет видимость модели, параметры, диагностические сценарии, минимальный расчет и регрессию.	Результат зафиксирован в TestReport; ошибки возвращены разработчику.
Release Manager	Принимает артефакты после QA и включает их в выпускной поток.	В выпуск попадает проверенный комплект, а не локальная отладочная копия.

3.4.4 Ключевые понятия

Понятие	Краткое объяснение
PRADIS	Расчетный комплекс, расширяемый пользовательскими моделями и объектами.
DINAMA	Каталог или среда поставки, где размещаются исполняемые библиотеки, XML-описания и служебные файлы.
Библиотека моделей	Группа моделей, поставляемая как общий исполняемый и описательный комплект.
Модель	Расчетный элемент, вызываемый решателем; формирует силовые вклады, рабочие величины и производные.
Объект	Пользовательский объект, связанный с XML-паспортом и, в предоставленных материалах, с Python-реализацией.
XML-паспорт	Файл описания библиотеки, модели или объекта для интерфейса: имя, порты, параметры, символ, описание.
DLL	Исполняемая библиотека с программной реализацией модели или набора моделей.
DEF-файл	Файл объявления экспортируемых процедур при сборке DLL.
sysarm.xml	Общий список разделов и библиотек, отображаемых в DINAMA.
plugin_repository.xml	Файл связи между паспортом модели, путем к DLL и именем вызываемой процедуры.
armdoc / arm	Инструменты регистрации, упомянутые в исходных материалах; точный синтаксис нужно сверить с целевой версией.

3.4.5 Состав комплекта поставки

Разработчик передает не один файл, а комплект, достаточный для сборки, регистрации, отображения, запуска и проверки модели. Неполный комплект приводит к тому, что модель может быть видна в интерфейсе, но не запускаться, или запускаться без воспроизводимого теста.

Компонент	Назначение	Что проверить
DLL	Скомпилированная библиотека моделей.	Имя и разрядность соответствуют PRADIS/DINAMA; путь совпадает с library.
Исходный код	Основа сопровождения и повторной сборки.	Код соответствует передаваемой DLL.
DEF-файл	Описание экспортируемых процедур.	Имена совпадают с procedure.
XML-паспорт библиотеки	Описание состава библиотеки.	Содержит все модели и объекты поставки.
XML-паспорт модели	Порты, параметры, рабочие переменные, символ, описание.	Согласован с кодом и использует корректную кодировку.
.for-описание	Служебное описание модели, упомянутое в исходных схемах.	Точная роль требует проверки для целевой версии.
Python-файл	Реализация объекта или Python-модели.	Путь размещения и API вызова подтверждены.
Справка	Поясняет назначение, параметры, ограничения и тест.	Пользователь понимает, когда применять модель.
Минимальный тест	Проверяет подключение и базовую физику модели.	Есть ожидаемый результат и диагностические сценарии.

3.4.6 Структура каталогов DINAMA

DINAMA/

plugin/ DLL пользовательских библиотек

plugin/python/pradis/<LibraryName>/ Python-реализации объектов или моделей

sysarm/

plugin_repository.xml связь модели, DLL и procedure

XML/

sysarm.xml общий список разделов и библиотек

<LibraryName>/

<LibraryName>.xml паспорт библиотеки

Model/<NameModel>.xml паспорт модели

Object/<NameObject>.xml паспорт объекта

Уточнение. Пути Python-плагинов, назначение .for-файлов, NEWOBJECT.sch, ПГО и ПРВП нужно сверить с актуальной поставкой. В этом руководстве они описаны только в объеме, подтвержденном предоставленными фрагментами.

3.4.7 Типовой маршрут: есть новая модель

1. Проверить постановку задачи: физический смысл, порты, параметры, единицы измерения, ограничения и ожидаемый результат.
2. Собрать или получить готовую DLL библиотеки модели либо подготовить Python-файл, если целевой механизм использует Python.
3. Подготовить XML-паспорт библиотеки и XML-паспорта моделей или объектов.
4. Разместить DLL в DINAMA/plugin; Python-файлы разместить в подтвержденном каталоге Python-плагинов.

5. Разместить паспорта в DINAMA/sysarm/XML/<LibraryName> и подпапках Model/Object.
6. Обновить sysarm.xml и выполнить регистрацию через armdoc/arm или утвержденный скрипт.
7. Проверить plugin_repository.xml: путь library, имя procedure, имя модели и фактическую DLL.
8. Открыть PRADIS/DINAMA и убедиться, что библиотека, модель, порты, параметры и символ отображаются корректно.
9. Запустить минимальный расчетный пример и диагностические сценарии.
10. Зафиксировать результат QA в TestReport или принятой внутренней форме.

3.4.8 Ручное включение библиотеки моделей

Ручной способ удобен при первичном подключении, локальной отладке и разборе ошибок автоматизации. Он показывает, какие файлы участвуют в процессе и какие связи должны появиться.

** Ис- полни- тель**	Действия
Разра- ботчик	Компилирует библиотеку; проверяет экспорт procedure; передает DLL, XML-паспорта, служебные описания, справку и минимальный тест.
Адми- нистра- тор	Копирует DLL в DINAMA/plugin; размещает XML в DINAMA/sysarm/XML/<LibraryName>; добавляет раздел библиотеки в sysarm.xml; обновляет или проверяет plugin_repository.xml.
QA	Проверяет видимость модели, соответствие параметров и портов паспорту, запуск минимального расчета и диагностические сообщения.

3.4.9 Автоматизированное включение библиотеки моделей

Автоматизированный способ использует armdoc/arm или заменяющие их скрипты. Он снижает объем ручной работы, но не отменяет проверку результата.

1. Сложить <LibraryName>.xml, <NameModel>.xml, служебные описания и сопутствующие файлы в общую рабочую папку.
2. Выполнить команды вида armdoc -a <LibraryName> и armdoc -a <NameModel> или утвержденный скрипт регистрации.
3. Проверить, что обновлены sysarm.xml, паспорт библиотеки и папка Model.
4. Выполнить команды вида arm + <NameModel> и arm ! <NameModel> или эквивалентную процедуру связи.
5. Проверить plugin_repository.xml и путь к DLL.
6. Запустить минимальную расчетную схему.

Важно. Если armdoc создал путь вида ../plugin/<NameModel>, а фактическая реализация находится в общей библиотеке NEWLIBRARY.dll, путь нужно заменить на ../plugin/NEWLIBRARY. Иначе PRADIS будет искать библиотеку по имени модели, а не по имени общей DLL.

3.4.10 XML-паспорта библиотеки, модели и объекта

XML-паспорт нужен для того, чтобы PRADIS/DINAMA мог показать пользователю библиотеку, модель, порты, параметры, описание и символ. Сам по себе паспорт не гарантирует запуск расчета: запуск зависит от корректной связи с исполняемым кодом.

Фрагмент	Назначение	Контроль
<module> </library>	/ Описание раздела или библиотеки.	Имя совпадает с каталогом <LibraryName>.
<model>	Отдельная расчетная модель.	name, module, image, ext, par, wrk согласованы с кодом.
<description>	Русское и английское описание.	Текст понятен пользователю и не противоречит справке.
<nodelist>/</node>	Порты модели.	Номера, имена и типы соответствуют реализации.
<parameterlist> </parameter>	Параметры модели.	Типы base.*, default и единицы согласованы с кодом.
<worklist>	Рабочие переменные.	Показаны только реально используемые величины.
<statelist>	Состояния модели.	Заполнен только при реальном использовании состояний.
<image2d>	Графический символ.	.PortSym соответствует номерам из nodelist.
<object>	Пользовательский объект.	Паспорт связан с Python-файлом или другим подтвержденным механизмом реализации.

3.4.11 plugin_repository.xml: связь модели с исполняемым кодом

plugin_repository.xml является ключевым местом связи между пользовательским описанием и исполняемым кодом. XML-паспорт может позволить интерфейсу показать модель, но расчет не запустится, если связь с DLL или procedure задана неверно.

<!-- Условный пример для проверки структуры, не нормативный XML-формат -->

```
<component model=»NameModel»>
```

```
<library>../plugin/NEWLBRARY</library>
```

```
<procedure>NAMEMODEL</procedure>
```

```
</component>
```

- library должен указывать на фактическую DLL без подмены имени модели на имя библиотеки.
- procedure должен совпадать с экспортируемой процедурой в DLL или с подтвержденным именем Python-адаптера.
- Имя модели в паспорте, имя записи в plugin_repository.xml и имя процедуры должны быть проверены как единая цепочка.
- После любой автоматической генерации plugin_repository.xml нужно открыть и проверить вручную или автоматическим тестом.

3.4.12 Добавление объекта

Объект отличается от расчетной модели тем, что в предоставленных материалах он связан не только с XML-описанием, но и с Python-реализацией. Поэтому проверяется пара: паспорт объекта и соответствующий .py-файл.

1. Подготовить <NameObject>.xml.
2. Подготовить <NameObject>.py или другой файл реализации, если целевой механизм отличается.
3. Разместить XML-паспорт в DINAMA/sysarm/XML/<LibraryName>/Object.
4. Добавить запись об объекте в паспорт библиотеки.
5. Разместить Python-файл в DINAMA/plugin/python/pradis/<LibraryName> или другом подтвержденном каталоге.
6. Проверить соответствие имени XML-паспорта, имени Python-файла и записи библиотеки.
7. Открыть PRADIS/DINAMA и выполнить минимальную проверку загрузки объекта.

Уточнение. В исходной схеме упоминается NEWOBJECT.sch и вопрос о генерации .py. Механизм генерации или связи нужно подтвердить отдельно перед включением в нормативную инструкцию.

3.4.13 Сквозной пример SV2K

SV2K используется как учебный пример связи паспорта с кодом. Модель описывает идеально упругую 2-D связь между двумя точками по поступательным степеням свободы. Параметры pA и pB задают начальные координаты, K задает коэффициент жесткости, а WRK хранит продольное усилие и начальную длину.

Данные паспорта		Фрагмент реализации	Смысл
Port1, Port2		X1, X2, X3, X4	Две точки по две поступательные степени свободы.
type=base.XY			
pA, pB		PAR(1)..PAR(4) PAR[0]..PAR[3]	/ Начальные координаты точек A и B.
K		PAR(5) / PAR[4]	Коэффициент жесткости.
wrk=2		WRK(1), WRK(2) / WRK[0], WRK[1]	Продольное усилие и сохраненная начальная длина.
Проверки параметров		NEWINT или initialize()	Однократная проверка длины и допустимости K.
Диагностика		CODE/NAME ModelStatus	или Сообщение о невозможности продолжить расчет.

$K = \text{PAR}(5)$

$U = LT - L$

$\text{WRK}(1) = K * U$

$I(1..4) = \text{проекция осевого усилия на степени свободы}$

$Y(1..16,1) = \text{производные сил по перемещениям}$

3.4.14 Разработка модели на Fortran

Обновлено. Fortran-раздел используется как базовый ориентир по расчетному контракту: массивы I, Y, X1..X4, PAR, WRK и COMMON-блоки связывают модель с вычислительным ядром.

Интерфейс

SUBROUTINE SV2K (I, Y, X1, X2, X3, X4, PAR, WRK)

** мент**	Аргу- мент	Назначение
I		Вектор силовых вкладов модели. Для 2D-связи используются четыре компоненты.
Y		Элементы якобиана, то есть производные сил по перемещениям.
X1, X2		Текущие перемещения точки A по X и Y.
X3, X4		Текущие перемещения точки B по X и Y.
PAR		Параметры модели: pA, pB и K.
WRK		Рабочий вектор для внутренних величин, которые нужно сохранить или показать.

Алгоритм

1. При NEWINT=1 проверить начальную длину и жесткость K.
2. Сохранить начальную длину L в WRK.
3. На каждом вызове вычислить текущие координаты точек A и B.
4. Найти текущую длину LT, направление связи, деформацию $U=LT-L$ и усилие $N=K*U$.
5. Записать силовые вклады в I и элементы якобиана в Y.
6. При вырождении длины вернуть диагностическое состояние через CODE и NAME.

Проверить. Состав COMMON-блоков приведен по учебному примеру. Перед выпуском официального шаблона нужно сверить актуальный состав переменных и соглашение вызова с целевой версией PRADIS.

3.4.15 Разработка модели на C

Новое. C-раздел показывает, как отразить Fortran-совместимый вызов в виде C-функции: указатели на массивы, нумерация с нуля, экспорт имени и доступ к COMMON-областям через внешние символы.

```
void SV2K_C(double *I, double *Y, double *X1, double *X2,
double *X3, double *X4, double *PAR, double *WRK);
```

Что проверить	Почему важно
Имя SV2K_C экспортируется из DLL	Иначе модель может отображаться, но не запускаться.
procedure совпадает с экспортом	Регистратор должен связать паспорт с фактической процедурой.
library указывает на нужную DLL	Иначе будет вызвана не та реализация или загрузка завершится ошибкой.
Разрядность DLL совпадает с PRADIS	32/64-битная несовместимость приводит к ошибке загрузки.
Индексация PAR/WRK переведена корректно	Fortran PAR(1) соответствует C PAR[0].
COMMON-структуры сверены	Ошибки выравнивания или имен символов могут повредить память.

LIBRARY NEWLIBRARY

EXPORTS

SV2K_C

Проверить. Соглашение вызова C ABI, регистр имени, нижние подчеркивания, cdecl/stdcall, имена COMMON-символов и упаковку структур нужно брать из официального C-шаблона или актуальной сборки PRADIS.

3.4.16 Разработка модели или объекта на Python

Уточнение. Python-реализация в исходных материалах разделена на расчетную часть и адаптер. Расчетная логика может тестироваться автономно, но точный API подключения к PRADIS/DINAMA должен быть подтвержден отдельно.

Часть	Назначение
S	Однократная проверка параметров и сохранение начальной длины.
V2KPy.initialize	
SV2KPy.calculate	Основной расчет сил, якобиана, рабочего вектора и статуса.
ModelStatus	Учебная форма диагностического результата: код, имя модели и сообщение.
sa	Предполагаемый адаптер, который должен быть заменен на официальный API
lculate(context)	PRADIS/DINAMA.
so	Условный источник пользовательских параметров pA, pB, K.
ncontext.parameters	
context.nodes	Условный источник текущих перемещений портов Port1 и Port2.

1. Подготовить Python-файл с расчетным классом и адаптером вызова.
2. Подготовить XML-паспорт модели или объекта.
3. Разместить Python-файл в каталоге, предусмотренном для пользовательских Python-объектов или Python-моделей.
4. Разместить XML-паспорт в соответствующем XML-каталоге DINAMA.
5. Обновить sysarm.xml или другую структуру регистрации, если это требуется целевой версией.
6. Проверить связь имени модели, Python-файла и XML-паспорта.
7. Запустить автономный тест расчетной логики и минимальный интеграционный тест PRADIS/DINAMA.

Проверить. В предоставленных материалах нет подтвержденной сигнатуры Python-модели PRADIS. Поэтому `calculate(context)`, `context.parameters`, `context.nodes` и `context.constants` являются учебным адаптером, а не нормативным API.

3.4.17 Тестирование после включения

Об- ласть	Что проверить
Интер- фейс	Библиотека видна в нужном разделе; модель отображается с корректным названием, описанием, портами, параметрами и символом.
Регис- тра- ция	<code>sysarm.xml</code> содержит раздел библиотеки; <code><LibraryName>.xml</code> содержит модель; <code>plugin_repository.xml</code> содержит корректные <code>library</code> и <code>procedure</code> .
Расчет	Минимальная расчетная схема строится и запускается; простой тест дает ожидаемый результат.
Диа- гности- ка	Некорректные параметры возвращают понятное диагностическое сообщение или код состояния.
Регрес- сия	Стандартные тесты PRADIS проходят после добавления новой библиотеки.

Минимальный тест SV2K

Параметр	Значение	Ожидаемый смысл
<code>pA</code>	(0, 0)	Начальная точка A.
<code>pB</code>	(1, 0)	Начальная точка B, начальная длина $L=1$ м.
<code>K</code>	1000 Н/м	Жесткость связи.
Перемещение B по X	0.001 м	Растяжение на 1 мм.
Ожидаемое усилие	около 1 Н	$N = K * U = 1000 * 0.001$.
Диагностика	$K < 0$ и нулевая длина	Модель возвращает диагностическое состояние.

3.4.18 Типовые ошибки и диагностика

Симптом	Вероятная причина	Что проверить
Модель не видна	Не обновлен sysarm.xml или паспорт библиотеки.	Раздел <LibraryName>, запись модели в <LibraryName>.xml.
Модель видна, но не запускается	Неверная связь с DLL.	plugin_repository.xml, library, procedure.
Ошибка загрузки DLL	DLL отсутствует или несовместима.	DINAMA/plugin, зависимости, разрядность, экспорт имен.
Параметры неверны	Ошибка XML-паспорта.	parameterlist, типы base.*, default, UTF-8.
Порты или символ неверны	Ошибка nodelist или image2d.	Номера портов, .PortSym, image.
Python-объект не найден	Не совпадают XML и .py или неверен путь.	Object/<Name>.xml и plugin/python/pradis/<LibraryName>/<Name>.py.
Расчет плохо сходится	Не заполнен или несогласован якобиан.	Y в Fortran/C или формат jacobian в Python API.
Диагностика не работает	Неверные CODE/NAME, COMMON или адаптер статуса.	COMMON-блоки, ModelStatus, обработку ошибок.

3.4.19 Процесс разработки, QA и выпуска

N	Исполнитель	Действие	Контроль
1	PM	Создает задачу и назначает разработчика.	Есть формальное задание.
2	Разработчик	Готовит код, паспорта, справку и тесты.	Комплект готов к проверке.
3	Разработчик	Передает изменения на QA.	QA получает воспроизводимый материал.
4	QA	Обновляет TestPlan и выполняет проверку.	Есть согласованный сценарий.
5	QA	Формирует TestReport.	Результат OK/NotOK зафиксирован.
6	Ответственный	При успешной проверке включает комплект в dev/выпускной поток.	Функциональность попадает в управляемую поставку.

Уточнение. Если в организации используется GitLab CI, Nexus, Redmine или иной pipeline, раздел нужно синхронизировать с фактическим процессом. Исходные материалы подтверждают общий маршрут: разработка, артефакты, QA, TestReport, решение о выпуске.

3.4.20 Требования к справке новой модели

- Назначение модели и область применения.
- Порты: имя, тип, физический смысл.
- Параметры: имя, тип, единицы, значение по умолчанию, ограничения.
- Рабочие переменные и состояния, если они доступны пользователю.
- Пример подключения в расчетной схеме.
- Ожидаемый результат и простой верификационный тест.
- Ограничения, особые случаи и диагностические сообщения.

- Версия библиотеки или дата редакции, если это принято в поставке.

3.4.21 Контрольные списки

Для разработчика

- Постановка задачи описана: физический смысл, параметры, ограничения.
- Сигнатура модели соответствует ожидаемому интерфейсу PRADIS.
- Порты и параметры XML-паспорта согласованы с индексами PAR или именами Python-параметров.
- Рабочий вектор WRK согласован с атрибутом wrk паспорта.
- В коде есть проверки начальной длины, допустимости K и других ограничений.
- Заполняются силовые вклады I и элементы якобиана Y, если они требуются расчетным API.
- Подготовлены XML-паспорт, справка, минимальный тест и ожидаемый результат.
- Для C проверены экспорт имени, ABI, разрядность и COMMON-структуры.
- Для Python подтверждены путь размещения и официальный API адаптера.

Для администратора

- DLL размещена в DINAMA/plugin.
- XML-паспорта размещены в DINAMA/sysarm/XML/<LibraryName>.
- Python-файлы, если они есть, размещены в подтвержденном каталоге.
- sysarm.xml содержит раздел библиотеки.
- plugin_repository.xml содержит корректный путь library и procedure.
- Автоматическая регистрация проверена вручную или тестом.

Для QA

- PRADIS/DINAMA запускается без ошибок загрузки новой библиотеки.
- Библиотека и модель видны в интерфейсе.
- Порты, параметры, значения по умолчанию и символ соответствуют паспорту.
- Минимальный расчет запускается и дает ожидаемый результат.
- Ошибочные параметры вызывают диагностическое состояние.
- Регрессионные тесты PRADIS не нарушены.
- Результат проверки зафиксирован.

3.4.22 Сведения, требующие уточнения перед нормативным выпуском

Тема	Что требует проверки	Влияние
Целевая версия PRADIS/DINAMA	Какая версия является целевой для руководства.	Критично для выпуска.
armdoc / arm	Официальный синтаксис команд и пример .bat-файла.	Критично для разделов регистрации.
plugin_repository.xml	Актуальный формат обязательных полей и связь с armctlg.	Критично для запуска модели.
.for-файл	Исходник, паспорт старого формата или вход для armdoc.	Желательно уточнить.
COMMON-блок	Актуальный состав переменных для целевой версии PRADIS.	Критично для Fortran/C шаблонов.
C ABI	Соглашение вызова, регистр имени, подчеркивания, cdecl/stdcall.	Критично для C-моделей.
Python API	Официальная сигнатура вызова Python-модели или объекта.	Критично для Python-раздела.
Каталоги Python	Где размещаются Python-файлы и их XML-паспорта.	Критично для установки.
NEWOBJECT.sch	Назначение и механизм связи с Python-объектами.	Желательно уточнить.
ПГО/ПРВП	Включать в это руководство или вынести отдельно.	Лучше вынести в отдельный документ.
Формат теста	Официальный формат минимальной расчетной схемы.	Желательно добавить в приложение.

3.4.23 ПРИЛОЖЕНИЕ - Пример структуры комплекта

Package/
 NEWLIBRARY.dll
 NewLibrary.cpp
 NewLibrary.def
 LibraryName.xml
 NameModel1.xml
 NameModel1.for
 NameModel2.xml
 NameModel2.for
 help/
 NameModel1.html
 tests/
 minimal_case/
 objects/
 NEWOBJECT.xml
 NEWOBJECT.py

DLL размещается в DINAMA/plugin. XML-паспорта библиотеки, моделей и объектов размещаются в DINAMA/sysarm/XML. Справка и тесты используются для проверки и сопровождения. Если объект реализован на Python, его .py-файл размещается в подтвержденном каталоге Python-плагинов библиотеки.

3.4.24 Глоссарий

Термин	Описание
ПГО	Постпроцессорный графический объект; в исходных материалах отмечен как продвинутая тема.
ПРВП	Пользовательская процедура вывода/постобработки; точная трактовка требует сверки с документацией.
TestPlan	План тестирования новой модели или библиотеки.
TestRepor	Отчет о результатах проверки.
Nexus	Хранилище артефактов сборки, если используется в процессе организации.
GitLab	Средство автоматической сборки и публикации артефактов, если используется в процессе организации.
CI	Система задач и фиксации статусов, если используется в процессе организации.
Redmine	Система задач и фиксации статусов, если используется в процессе организации.