



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 104 за 2014 г.



Горбунов-Посадов М.М.,
Ролдугин Д.С., Слепенков М.И.,
Тузов И.В.

Анимация и видео в научной
публикации

Рекомендуемая форма библиографической ссылки: Анимация и видео в научной публикации / М.М.Горбунов-Посадов [и др.] // Препринты ИПМ им. М.В.Келдыша. 2014. № 104. 32 с. URL: <http://library.keldysh.ru/preprint.asp?id=2014-104>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

**М.М.Горбунов-Посадов, Д.С.Ролдугин,
М.И.Слепенков, И.В.Тузов**

**Анимация и видео
в научной публикации**

Москва – 2014

Горбунов-Посадов М.М., Ролдугин Д.С., Слепенков М.И., Тузов И.В.

Анимация и видео в научной публикации

Рассматриваются характерные примеры включения мультимедийных возможностей в научную публикацию: авторская видео аннотация в карточке публикации, встроенные в файл формата PDF анимированные и трехмерные иллюстрации.

Ключевые слова: научная публикация, видео, анимация, 3D

***Mikhail Mikhailovich Gorbunov-Posadov, Dmitry Sergeevich Roldugin,
Mikhail Ivanovich Slepencov, Ilya Vladimirovich Tuzov***

Animation and video in scientific publication

The typical examples of multimedia in scientific publication are considered, such as the author's video abstract in the ID-card of the article and also animated and 3D illustrations built in PDF-file.

Key words: scientific publication, video, animation, 3D

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 13-01-00493-а.

Оглавление

Введение	3
Видео аннотация	4
Подготовка файла видео	6
Способы включения видео в PDF-документ	9
Добавление видео в PDF средствами Adobe Acrobat	10
Поддержка видео WEB-браузерами	13
Поддержка видео мобильными платформами	16
Встраивание видео при помощи LaTeX.....	17
Встраивание потокового видео – YouTube.....	21
Трехмерная иллюстрация	23
Заключение.....	30
Литература	31

Введение

Интернет уверенно превращается в основной источник научного знания. Научные журналы, монографии, энциклопедии либо получают полнотекстовую интернет-проекцию, на которую с удовольствием переключается подавляющее большинство читателей, либо и вовсе целиком переключиваются в интернет, постепенно забывая о своих печатных корнях.

Среди многочисленных преимуществ размещения научной публикации в интернете далеко не последнюю роль играет возможность широко использовать звук, видео, анимацию, интерактивную компьютерную графику – богатейший набор средств, о которых не могли и мечтать печатные издания. К сожалению, эти средства мультимедиа включаются в научные публикации недостаточно часто.

А ведь технический арсенал мультимедиа учеными давно освоен. Редкий семинар или защита диссертации проходят сегодня без подготовленной докладчиком динамичной демонстрации сталкивающихся галактик, движущихся циклонов, совершающих головокружительные манипуляции роботов и т. д. Почему же эти замечательные иллюстрации до сих пор не украшают собой публикуемые научные работы?

Причин несколько. Прежде всего, инерция представления о публикации как о чем-то абсолютно неподвижном, что обязано в полном объеме отразиться на бумаге.

Показательно в этом плане отношение к мультимедиа в ВАК. С одной стороны, в ВАК недавно совершен решительный поворот лицом к интернету. И полный текст диссертации, и отзывы оппонентов, и многие другие материалы защиты размещаются в свободном доступе на сайте диссертационного совета. Но с другой стороны, диссертант, хотя и с удовольствием демонстрирует презентацию с мультимедиа на защите, тем не менее не решается включить в свой автореферат и диссертацию видео или анимацию. Ведь наряду с онлайн-версией по старинке готовится печатная версия диссертационного материала, и тут, по-видимому, молчаливо предполагается, что обе версии должны оказаться строго тождественными.

Печальное недоразумение. Давно пора в регламентирующих документах ВАК закрепить ведущую роль онлайн-версии диссертации и автореферата, явным образом разрешив там включение мультимедиа. В таком случае изредка еще встречающийся любитель ретро – читатель печатной версии – получит то же, что и раньше, хотя и испытает некоторое неудобство. Но зато как выиграет массовый онлайн-читатель!

Сходная картина наблюдается и в большинстве научных журналов. Чрезвычайно редко автору статьи разрешается подготовить мультимедиа иллюстрации для онлайн-версии. Как ни странно, даже если журнал исключительно онлайн-версии, тем не менее мультимедийные иллюстрации в нем зачастую не допускаются.

Помимо упомянутой инерции печатных изданий, существуют и чисто технические причины, мешающие массовому появлению мультимедиа иллюстраций в научной статье. В то время как у создателя "ненаучного" сайта с появлением HTML5 просто глаза разбегаются от обилия всевозможных доступных приемов работы с мультимедиа, обустройство "научного" интернета вызывает определенные сложности. Дело в том, что для представления текстов научных статей HTML практически не используется, там безраздельно господствует формат PDF. А организация размещения мультимедиа в PDF коренным образом отличается от технологий HTML, причем, увы, не в лучшую сторону.

Какой путь должна пройти мультимедийная иллюстрация до появления в онлайн-научной статье? Прежде всего отметим, что непосредственно в формате PDF статья не создается. Текст статьи готовится обычно в MS Word или в TeX, а мультимедийные иллюстрации – в специализированных средах типа Matlab. Далее предстоит соединить текст и мультимедиа, и тут возникают определенные сложности.

Прежде всего, надо выбрать последовательность действий. Либо попытаться встроить мультимедиа в Word или TeX, а затем преобразовать полученный файл в PDF, либо сначала преобразовать Word или TeX (без мультимедиа) в PDF, а затем встраивать мультимедиа непосредственно в PDF.

И Word, и TeX, каждый по-своему, сопротивляются вставке мультимедиа. Но даже если это удалось проделать, радоваться рано: при последующем преобразовании полученного файла в PDF мультимедийные возможности иллюстраций могут быть утрачены. Встраивание же мультимедиа непосредственно в PDF требует привлечения не всегда имеющихся под рукой инструментальных средств.

Тем не менее, как будет показано в последующих разделах, задача встраивания мультимедийных иллюстраций в онлайн-научную PDF-публикацию вполне разрешима и после выработки определенных навыков не составляет особого труда.

Однако рассмотрение роли мультимедиа и приемов размещения мы начнем с относительно простого HTML-примера.

Видео аннотация

Научная публикация обычно атрибутируется определенным набором метаданных: авторы, название, аннотация, ключевые слова, год издания, число страниц и др. Часто оказывается полезным дополнить этот набор видео аннотацией, где автор, разумеется, не зачитывает механически текст традиционной аннотации, а формулирует свое отношение к предмету публикации, стараясь установить контакт с читателем, привлечь его внимание к работе в целом и к отдельным ее наиболее важным аспектам. Наряду с непосредственной записью обращения автора к читателю в качестве видео аннотации может выступать и небольшой анимированный ролик,

иллюстрирующий наиболее увлекательные визуальные компоненты публикации.

При онлайн-овом размещении публикации ее метаданные, как правило, выносятся на отдельную страницу, представляемую в формате HTML. В таком случае на этой же странице следует, по-видимому, разместить и видео аннотацию (рис. 1).

The screenshot shows a web browser window with the address bar containing the URL <http://library.keldysh.ru/preprint.asp?id=2013-51>. The page title is "Аннотация публикации". The main heading is "Публикация". Below the heading, the text reads: "Препринт ИПМ № 51, Москва, 2013 г." and "Авторы: **Малинецкий Г.Г.**". The title of the preprint is "Синергетика, междисциплинарность и постнеклассическая наука XXI века". The annotation text discusses the philosophy of science and interdisciplinary approaches. To the right of the text is a video player showing a man in a suit speaking. Below the video player are the YouTube logo and the text "YouTube".

Электронная библиотека • Поиск публикаций English

Публикация

Препринт ИПМ № 51, Москва, 2013 г.

Авторы: **Малинецкий Г.Г.**

Синергетика, междисциплинарность и постнеклассическая наука XXI века

Аннотация:

Рассматривается направление философии науки, связанное с прогнозом развития основных тенденций познания. Предложенный взгляд опирается на междисциплинарные подходы, играющие важнейшую роль в современной науке, на логико-методологический анализ и теорию типов рациональности, выдвинутую В.С.Стёпиным, на идеи постнеклассики. Показано, что в точке бифуркации, которую сейчас проходит познание, возникают принципиальные трудности в продолжении ряда исследовательских программ, однако появляются новые возможности и перспективы. Рассмотрены ограничения в сфере науки, которые все глубже осознаются исследователями, а также быстрое и глубокое изменение самого субъекта познания, ожидающее нас в ближайшие десятилетия.

Ключевые слова: философия науки, типы рациональности, междисциплинарные подходы, когнитивный барьер, внутринаучная рефлексия, синергетика, постнеклассика, математическая история, большие данные, стратегический прогноз, обратные задачи, горизонт прогноза, внутринаучная рефлексия

Язык публикации: русский, страниц: 36

Направление исследований: **Математическое моделирование в актуальных проблемах науки и техники**

Полный текст: http://keldysh.ru/papers/2013/prep2013_51.pdf

Сведения об авторах:

- Малинецкий Георгий Геннадьевич, gmalin@Keldysh.ru, ИПМ им. М.В. Келдыша РАН

Рис. 1. Видео аннотация на странице метаданных <http://library.keldysh.ru/preprint.asp?id=2013-51>

Как уже упоминалось, размещение элементов видео в HTML трудностей не вызывает. На рис. 1 представлено одно из наиболее простых и экономичных решений – использование сервиса YouTube.

С точки зрения автора, видео аннотация вообще не несет в себе никаких проблем: от автора требуется лишь доставить свой видео файл в редакцию, а размещением его на странице метаданных занимается персонал веб-издания. Заметно больше усилий потребуется от автора, если мультимедийный материал непосредственно в форме иллюстрации вставляется в текст публикации. Некоторым способам реализации таких иллюстраций посвящены последующие разделы.

Подготовка файла видео

В форме анимированной графики или видеофрагмента часто представляются материалы и результаты исследований, описывающие динамические процессы и сложные объекты. Распространенными источниками видео для научной публикации являются:

- математические и вычислительные пакеты: Matlab, Mathematica;
- инструменты конечно-элементного анализа, в том числе моделирования физических процессов (аэродинамики, механики и пр.): ANSYS Fluent, ANSYS Mechanical, Abaqus;
- инструменты 3D-моделирования и рендеринга: 3ds Max и аналоги;
- CAD/CAM системы: CATIA, Autocad, MentorGraphics PCB;
- инструменты визуализации научных расчетов: Tecplot;
- непосредственная видеозапись эксперимента/объекта исследования.

Подготовка видеофрагмента для включения в публикацию состоит из следующих шагов.

1. Формирование файла исходного видеоматериала. На этом этапе важно правильно выбрать разрешение видеофрагмента. Низкое разрешение доставляет дискомфорт читателю и мешает восприятию, однако слишком высокое разрешение существенно увеличит размер PDF-документа и будет вызывать зависания reader`а при перелистывании и запуске видео для просмотра. Тем не менее, высокое разрешение исходного файла здесь предпочтительнее, так как его можно будет затем уменьшить на этапе конвертирования.

2. Опциональная доработка видеофрагмента в редакторе для повышения наглядности. Выделение в видеофайле наиболее существенного/значимого материала и перемещение его в начало видеофрагмента, формирование замещающего рисунка ("постера"), добавление титров.

3. Конвертирование исходного файла в формат с высокой компрессией. В ранних версиях Adobe Reader поддерживались различные кодеки и видеоконтейнеры, но начиная с версии X Adobe поддерживает только видео, закодированное в формате h264 в контейнерах mov, mp4, m4v, flv, f4v и swf. Поскольку кодеки стандарта h264 на данный момент позволяют получить наиболее высокую степень компрессии, данные требования представляются вполне оправданными.

Примечание. В то же время перспективный стандарт кодирования h265 по заявленным возможностям обещает обеспечить вдвое лучшую компрессию по сравнению с h264/MPEG-4 AVC. Появление более эффективных кодеков позволяет уменьшать объем PDF-документов со встроенным видео или же, соответственно, повышать разрешение/частоту кадров.

В качестве примера сформируем видеоролик в пакете Matlab и подготовим его для включения в PDF конвертером ffmpeg. Построение и анимация трехмерной поверхности выполняется следующим фрагментом кода на языке Matlab [1].

```
clear; close all;
% 2-мерная сетка для построения поверхности
x=linspace(0,1,100);
[X,Y] = meshgrid(x,x);
set(gcf, 'PaperPositionMode', 'auto');
set(gcf, 'Position', [200 200 800 600]);
N=240; % количество кадров
for i = 1:N
    % поверхность XYZ в зависимости от N
    Z = sin(2*pi*(X-i/N)).*sin(2*pi*(Y-i/N));
    surf(X,Y,Z);
    set(gcf, 'color', 'w');
    % сохранение кадра
    M(i)=getframe(gcf);
end
% Формирование видео как avi файла без сжатия 24 кадра/сек
movie2avi(M, 'WaveMovie.avi', 'compression', 'None', 'fps', 24);
```

Для каждого значения $i \in [1:N]$ формируется один кадр изображения, который затем сохраняется в элементе массива $M(i)$. Данный массив кадров передается в качестве первого аргумента функции `movie2avi(mov, filename, ParameterName, ParameterValue)`. Вторым аргументом является имя формируемого файла видео. Далее могут быть добавлены пары ", Имя параметра, Значение параметра". Парами имя-значение можно задать коэффициент компрессии, количество кадров в секунду, количество опорных кадров, качество изображения (от 0 до 100). Более подробно с данной функцией можно ознакомиться в руководстве Matlab [2].

Захват видео происходит непосредственно из окна Plot, поэтому разрешение результирующего файла может быть задано за счет настройки области отображения текущей фигуры, указатель на которую содержится в current figure handle – `gcf`. Вызов функции `set(gcf, 'Position', [200 200 800 600])` задает нижний левый угол в точке (200, 200) от нижнего левого угла экрана, размер задается равным 800 точек по ширине и 600 точек по высоте. Сформированный видеоролик `WaveMovie.avi` будет помещен в корневую директорию проекта. В приведенном примере получен файл видео длительностью 10 секунд – исходные 240 кадров с частотой 24 кадра в секунду и разрешением 800x600. Доработки в редакторе данное видео не требует.

Размер сформированного файла видео велик – 329 Мб, и было бы неразумно включить его непосредственно в PDF. Поэтому конвертируем его в формат `mp4` со сжатием кодеком `h264` при помощи open-source конвертера `ffmpeg`. Для данной программы существуют графические оболочки, но полная функциональность доступна только в режиме командной строки [3].

В строке аргументов `ffmpeg` ключ `-i` задает входной файл, ключ `-r` – частоту кадров, ключ `-vcodec` – настройки кодека. В настройках можно также указать координаты и размер фрагмента, длительность, частоту кадров, разрешение и пр. Например, если в качестве исходного материала имеется последовательность кадров (файлов изображений), то можно получить из них файл видео командой: `ffmpeg -r 25 -y -i "image_%010d.png" output.mpg`.

В данном случае выбираем кодек `h264`, результат помещаем в файл `Wave.mp4`. Зайдем в каталог с файлом видео: `C:\>cd C:\codec\`. И выполним следующую команду:

```
C:\codec>ffmpeg -i WaveMovie.avi -vcodec libx264
-pix_fmt yuv420p Wave.mp4
```

В процессе работы конвертер формирует лог со служебной информацией, где в том числе представлено количество обработанных кадров, средняя скорость потока (`bitrate`), ключевые кадры.

Данный формат команды является минимально необходимым. Важно указать формат пиксела именно `-pix_fmt yuv420p`, т.е. Plain YUV 4:2:0. Как показывает практика, при других форматах видео в PDF будет отображаться некорректно или не будет отображаться вовсе. Для точного позиционирования в пределах файла видео рекомендуется добавлять необходимое количество опорных/ключевых кадров, для чего служит ключ *-g кол-во ключевых кадров в секунду*.

Полученный файл имеет размер 6,9 Мб при тех же частоте кадров и разрешении, т.е. коэффициент сжатия составил 47. Для того чтобы оценить, достаточно ли выполненного сжатия для включения файла в публикацию, можно конвертировать все требуемые файлы видео в формат `h264`, вычислить их суммарный объем и сопоставить его с ограничением на размер итогового файла PDF. Обычно считается, что размер PDF-документа не должен превышать 20 Мб с учетом сегодняшних ограничений Web-сервисов и разумного времени загрузки. Однако сервисы, размещающие публикации, могут устанавливать и более жесткие требования.

Дальнейшее сокращение размера файла рекомендуется выполнять за счет уменьшения разрешения. Например, уменьшим разрешение исходного файла на 25% по ширине и высоте: `600x450`. Для этого выполним следующую команду `ffmpeg`:

```
C:\codec>ffmpeg -i WaveMovie.avi -s 600x450
-vcodec libx264 -pix_fmt yuv420p Wave_final.mp4
```

Полученный файл `Wave_final.mp4` имеет размер 3,15 Мб, т.е. размер файла сократился более чем в два раза. Теперь файл можно включить в PDF-документ.

Также для размещения видео в документе необходимо сформировать статическое изображение ("постер"), отображаемое на месте видеофрагмента до

его загрузки, запуска или в случае, когда видео не удастся запустить. Статическое изображение проще всего получить непосредственно из среды моделирования (Matlab в данном случае), но если исходный проект не доступен, то можно получить постер из готового видеофрагмента при помощи `ffmpeg`, выполнив команду:

```
c:\codec>ffmpeg -i Wave.mp4 -r 1 image%03d.jpg
```

Данная команда извлекает изображения из исходного видео – 1 кадр для каждой секунды файла видео. Выберем один из полученных файлов в качестве постера.

Способы включения видео в PDF-документ

Видеофрагмент можно включить в PDF-документ двумя способами:

- прикрепить файл видео непосредственно к документу;
- загрузить на видеохостинг (YouTube и т.п.) и включить в PDF-документ plug-in плеер, воспроизводящий видеопоток с данного хостинга.

В первом случае основным преимуществом является независимость от подключения к интернету и от ошибок видео хостинга. Очевидный недостаток – существенное увеличение размера PDF-файла. Поскольку обычно существуют ограничения на размер файла для размещаемых в сети документов, то соответствующие ограничения автоматически проецируются и на встраиваемое видео, т.е. тут возможно придется уменьшать разрешение и/или длительность.

Во втором случае преимуществом является незначительный размер блока кода, добавляемого непосредственно в файл документа, благодаря чему открывается возможность включить в документ практически неограниченное количество видеофрагментов высокого качества. Все возникающие здесь ограничения на видеоконтент зависят только от выбранного видеохостинга. Недостатком данного способа является полная зависимость от хостинга, предоставившего ссылку, в частности, от его потенциальных ошибок: видео может оказаться недоступно или, что еще хуже, по данной ссылке может неожиданно появиться какой-либо другой видеоролик.

При размещении на видеохостингах возникают и другие сложности. YouTube, например, оптимизирует загруженное видео в соответствии со своими правилами, которые зачастую неизвестны пользователю. Видео может быть конвертировано с существенными потерями в качестве, часто изменяется разрешение без сохранения пропорций. Кроме того, в качестве статического рисунка ("постера") можно установить только одно из трех изображений, которые сервис выделяет из загруженного видео, а качество (разрешение) такого изображения, как правило, очень низкое.

Итак, небольшие видеофрагменты удобно включать непосредственно в PDF-документ, что позволит получить самодостаточный PDF – с возможностью просматривать интегрированное видео без подключения к сети, с минимальной задержкой при запуске, без зависаний при просмотре. Однако нередко встречаются ситуации, когда целесообразным становится обращение к хостингу:

- требуемое уменьшение размера файла вызывает недопустимое или нежелательное ухудшение качества видеопотока;
- требуется включить множество видеофрагментов, недопустимо увеличивающих размер публикации;
- PDF-публикация ориентирована исключительно на интернет-аудиторию, т.е. будет просматриваться на устройствах со стабильным доступом к сети;
- видео на хостинге является самодостаточным объектом и служит не только для целей данной PDF-публикации; например, видеоролик существует на хостинге уже продолжительное время и, возможно, имеет высокий рейтинг и значительное количество комментариев.

Для подготовки научных публикаций используются различные средства: WYSIWYG-редакторы (Microsoft Word, OpenOffice Writer и т.п.) или системы компьютерной верстки LaTeX.

Microsoft Word допускает включение видео в документ. Однако при сохранении документа в формате PDF возможность просмотра видео утрачивается, на месте видео объекта остается лишь изображение или вовсе черный фон. Поэтому в данном случае рекомендуется при верстке документа на месте предполагаемых видеофрагментов поместить статическое изображение – "постер", а после экспорта в PDF отредактировать документ в Adobe Acrobat, заменив изображение на объект видео, т.е. включать видео непосредственно в PDF.

При верстке документа в LaTeX проблема существенно упрощается, так как здесь для включения мультимедийных объектов существует специальный пакет `media9`. Здесь видео включается в документ TeX специальной командой, реализованной в пакете `media9` с возможностью детальной настройки, и после компиляции в PDF документ обладает полной видео функциональностью без необходимости доработки его в редакторе Acrobat.

Добавление видео в PDF средствами Adobe Acrobat

Как упоминалось выше, при верстке документа в Microsoft Word включать видео рекомендуется после его экспорта/сохранения в формате PDF. Затем с PDF работает Adobe Acrobat, где для включения видео надо будет выполнить следующую последовательность действий.

1. Открыть сохраненный PDF-документ для редактирования в Acrobat. Найти картинку, резервирующую место под видео-объект, и удалить ее при помощи инструмента "Изменить объект" из меню "Инструменты → Контент".
2. Добавить видео-объект. В том же меню "Контент" выбрать инструмент "Мультимедиа → Видео..." и задать для размещения видео прямоугольную область на странице (размеры и положение которой позднее можно будет откорректировать). В открывшемся окне "Вставить видео" (рис. 2) выбрать файл видео (кнопка "Обзор..."). Начиная с версии X допустимыми форматами видео являются: flv, f4v, mov, mp4, m4v, 3gp, 3g2. Как уже было сказано, рекомендуется сохранять видео в формате mp4, закодированным h264-кодеком.
3. Настроить параметры. Стиль воспроизведения: на странице или в отдельном окне. Преимуществом отдельного окна является возможность изменения размера; также можно перелистывать страницы документа, при этом видео будет по-прежнему отображаться, так как окно просмотра видео не будет привязано к конкретной странице. Разрешение (ширина и высота) рекомендуется установить исходное, т.е. разрешение, которое указано в параметрах файла, 600x450 в нашем примере.
4. Добавить постер (замещающее изображение) – файл со статическим изображением, полученным ранее, например, утилитой ffmpeg.
5. Добавить кнопки и скрипты (опционально).

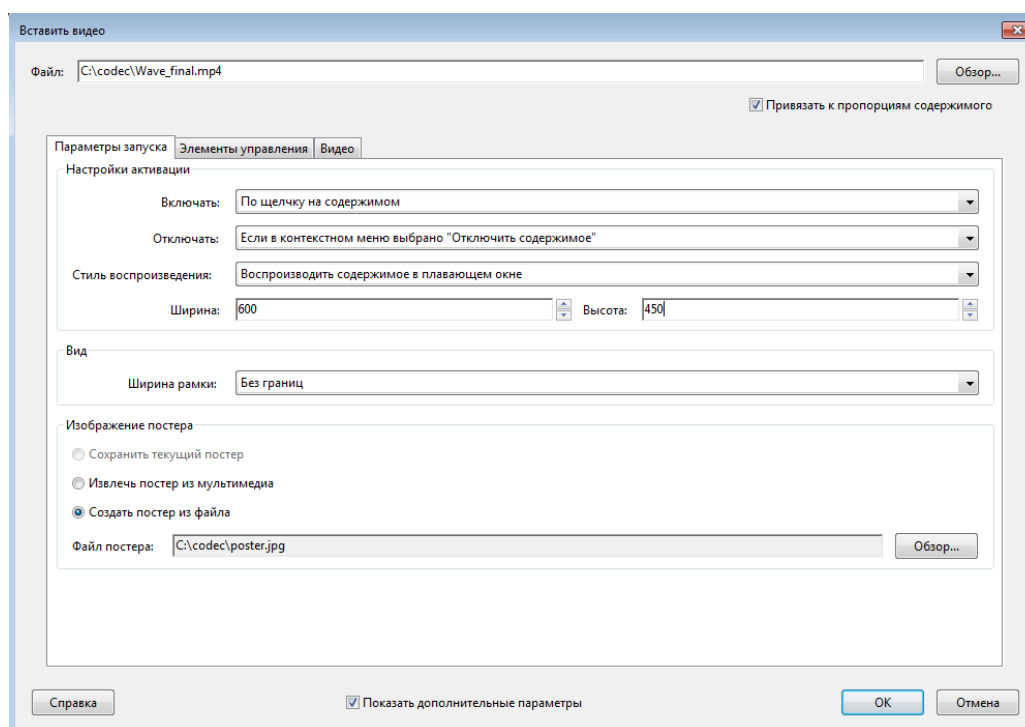


Рис. 2. Добавление видео в Adobe Acrobat

Для того чтобы читатель мог отличать статические картинки от видео, можно добавлять подсказку в подрисуючную надпись. Более выразительный

способ подсказки, расширяющий функциональность для читателя – добавление кнопок, которые активируют различные действия с видео: запуск, пауза, остановка, перемотка и пр. Добавить кнопку можно одноименным инструментом из меню "Инструменты → Формы" в Acrobat до версии 9 включительно или из меню "Инструменты → Контент" в Acrobat версий X и старше. В открывшемся окне свойств можно добавить подсказку (Tooltip) и настроить внешний вид кнопки. Заменить стандартный вид кнопки на изображение из файла можно во вкладке "Параметры": выбрать макет "Только пиктограмма" и добавить изображение из файла – кнопка "Выбор пиктограммы". Рекомендуется использовать интуитивно понятные изображения и прозрачный фон.

Настроить действия, активирующиеся кнопкой, можно в меню "Действия → Операция с мультимедиа" (Acrobat 9 и выше): нажать кнопку "Добавить" (рис. 3). Выбрать интересующую операцию (воспроизведение/пауза/перемотка и т.п.) и аннотацию к мультимедиа – видео-объект, с которым связано данное действие. Здесь же можно добавить собственный JavaScript-сценарий.

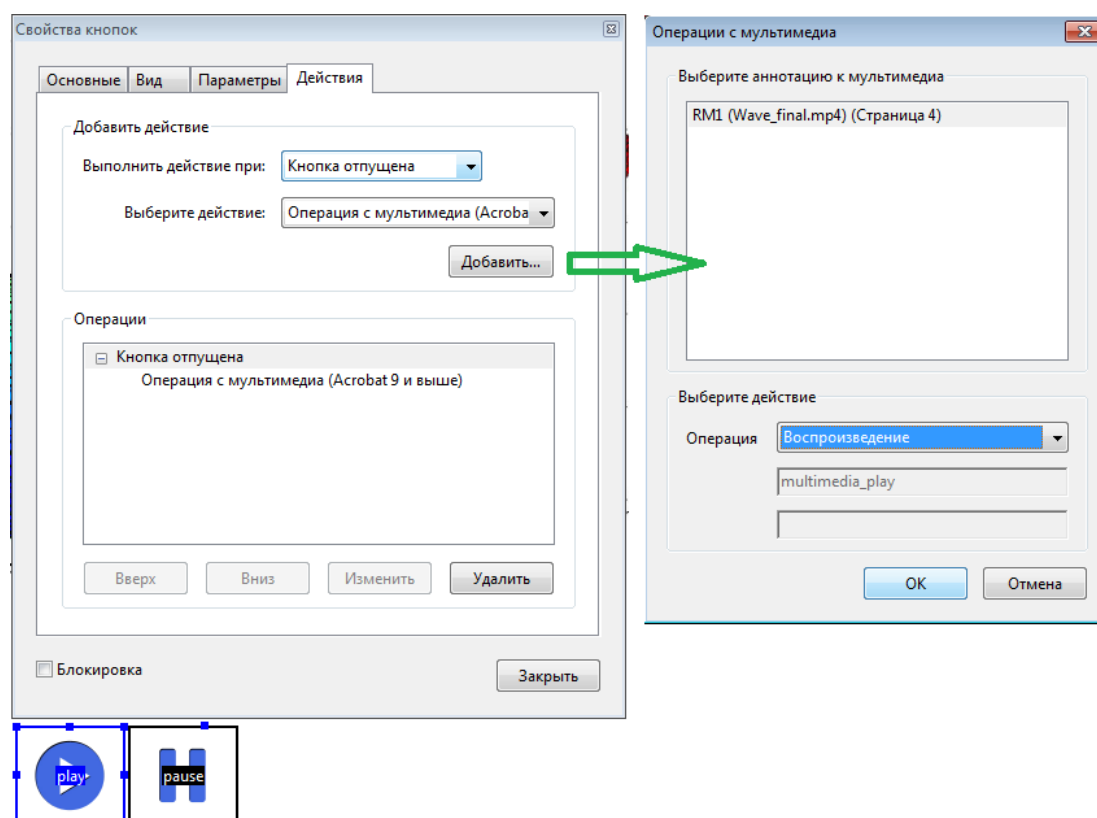


Рис. 3. Настройка действий, активируемых кнопкой

Стоит отметить, что в Acrobat до версии 8.1 включительно при добавлении видео были доступны дополнительные настройки и возможности. Одна из полезных возможностей – доступ к видео из JavaScript-кода через объект `RenditionObject`. В руководстве по JavaScript для Acrobat [4] содержится множество примеров для управления воспроизведением видео при помощи

JavaScript-кода: настройка параметров воспроизведения и запуска, выбор плееров, остановка, перемотка, доступ к открытым плеерам, перехват ошибок и т.п. С видео связывался один или несколько объектов `RenditionObject`, задающих параметры воспроизведения.

Например, используя объект `Rendition` с именем `WaveRendition`, следующий JavaScript-код запускает плеер для воспроизведения видео `WaveAnimation` в полноэкранном режиме:

```
app.media.openPlayer({
  rendition: this.media.getRendition( "WaveRendition" ),
  annot: this.media.getAnnot(
    {nPage:0,cAnnotTitle:"WaveAnimation"}),
  settings: { windowType: app.media.windowType.fullScreen }
});
```

А следующий JavaScript сценарий закрывает все активные плееры:

```
var players = media.getOpenPlayers();
for (var i in players)
  players[i].close( app.media.closeReason.general );
```

В последних версиях Acrobat доступ к видео осуществляется через объект `RichMedia`, получить который можно функцией `getAnnotsRichMedia(номер страницы)[номер медиа объекта в массиве]`. Закрыть активные плееры можно следующим сценарием:

```
rm = this.getAnnotsRichMedia(this.pageNum);
for(var i in rm)
  if(rm[i].activated)rm[i].activated = false;
```

Назначим данный сценарий на дополнительную кнопку. Тем самым у читателя появится возможность закрывать воспроизведение видео удобным способом.

Сформированная таким образом видео аннотация представлена на рис. 4 – на странице отображается замещающее изображение, при щелчке мыши на котором активируется видео в отдельном окне. Кнопки управления рядом с видео аннотацией подсказывают читателю, что объект является интерактивным, и одновременно помогают запускать, останавливать и закрывать видео.

Поддержка видео WEB-браузерами

Современные web-браузеры позволяют просматривать непосредственно в своей среде не только HTML-страницы, но и документы некоторых других типов, в частности, PDF-документы. Однако читателя здесь нередко ожидает серьезная неприятность: встроенные в браузер средства просмотра PDF не поддерживают мультимедийные объекты. На месте видео читатель увидит только статическую картинку ("постер") и зачастую даже не получит уведомления о том, что в просматриваемом документе содержится мультимедийные иллюстрации, которые не поддерживаются браузером.

Полноценно показать мультимедиа мог бы плагин Adobe Reader, обладающий всей необходимой функциональностью для просмотра PDF, но даже если он установлен, то по умолчанию в большинстве браузеров он не активирован.

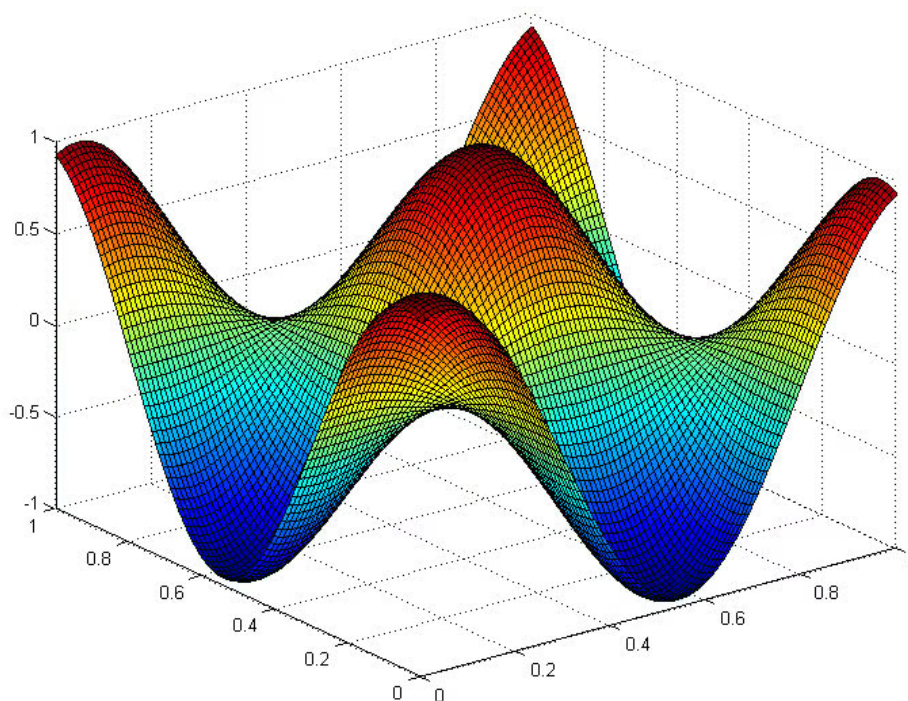


Рис. 4. Видеофрагмент, построенный в Matlab, включен в PDF при помощи Acrobat

Разумеется, читатель может самостоятельно открыть публикацию в полнофункциональном reader`е или активировать Adobe Reader plug-in в браузере. Например, в браузере Google Chrome для этого нужно открыть страницу расширений – ввести в поле адреса `chrome://plugins`, отключить там Chrome PDF Viewer и включить Adobe Reader (рис. 5). После этого проблем с просмотром встроенного мультимедиа не возникнет.

Однако догадается об этом далеко не каждый. Читателю необходимо явно подсказать, что используемый браузером reader не поддерживает мультимедиа. И поскольку браузеры чаще всего не выдают соответствующего предупреждения, данная задача ложится на плечи автора публикации. Решить ее в большинстве случаев позволяет следующий несложный JavaScript-сценарий, встраиваемый в PDF-документ:

```
if(!app.media) {
    app.alert("Приложение не поддерживает мультимедиа-объекты.
    Для просмотра данного документа рекомендуется
    использовать Abobe Reader/Plugin");
}
```

```

}
var mp = app.media.getPlayers();
if(mp == null){
    app.alert("Приложение не поддерживает воспроизведение
    встроенных видео-объектов. Для просмотра данного
    документа рекомендуется использовать
    Abobe Reader/Plugin");
}

```

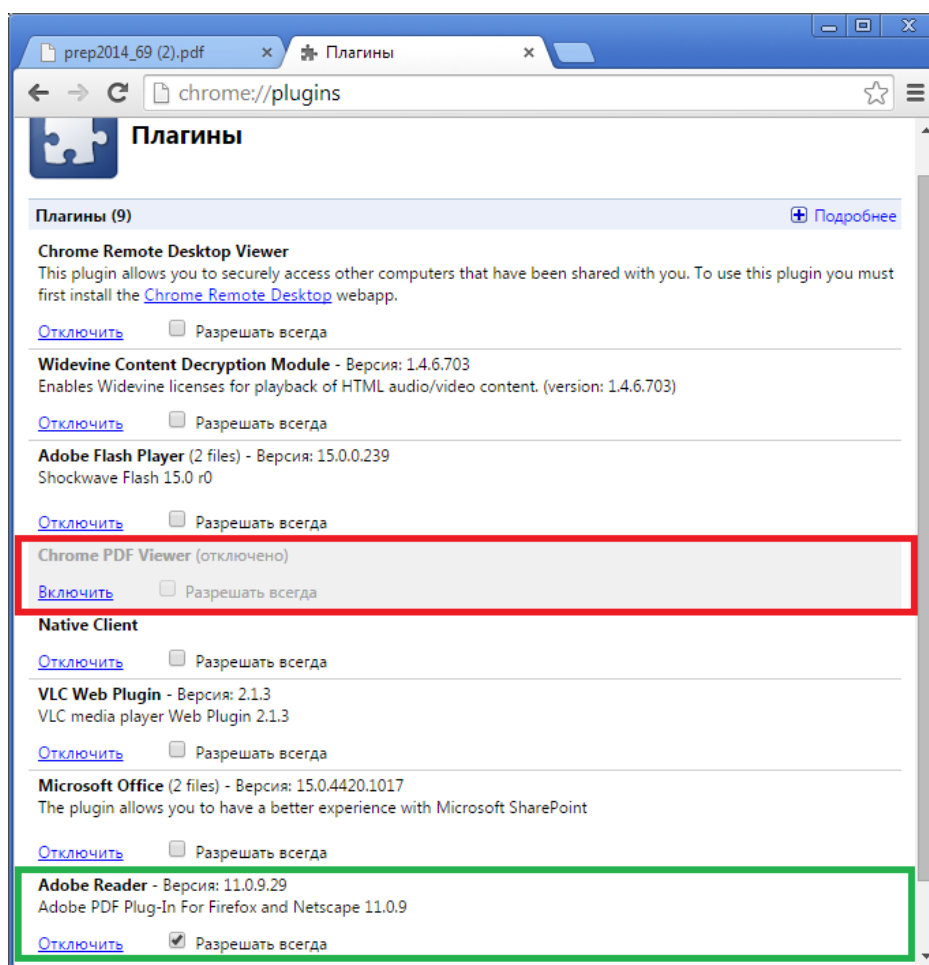


Рис. 5. Включение в Google Chrome расширения Adobe Reader

Скрипт пытается получить доступ к объекту `app.media` (мультимедиа). Если объект недоступен, то это означает, что данная программа просмотра документа PDF не поддерживает мультимедиа, в этом случае читатель видит предупреждение (рис. 6). Далее проверяется возможность получить доступ к видеоплееру и выдается соответствующая диагностика.

Данный сценарий назначается на два события: первое происходит сразу после загрузки документа, второе – при попытке запуска видео (при нажатии на кнопку play). Тем самым и при первоначальной загрузке документа, и при нажатии на кнопку play сценарий определит возможность просмотра видео и, если надо, подскажет читателю приложение, подходящее для просмотра.

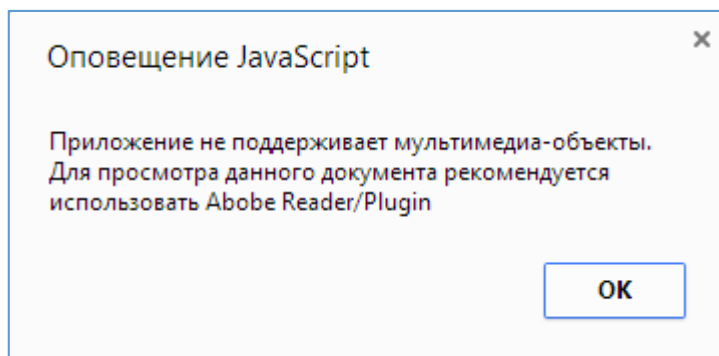


Рис. 6. Предупреждение в web-браузере, сформированное сценарием JavaScript

Поддержка видео мобильными платформами

В связи с массовым перемещением читателей на мобильные платформы Android и iOS стоит отдельно упомянуть об ограничениях этих операционных систем в части мультимедийных иллюстраций в PDF-документах. Ни Android, ни iOS не поддерживают технологию Adobe Flash, с которой неразрывно связано встраивание мультимедийных объектов в PDF, начиная с 9-й версии Adobe Acrobat.

Только desktop-версия Adobe Reader может воспроизводить мультимедийное содержимое, созданное по технологии Flash (swf, flw) или совместимое со стандартом H.264, для чего требуется подключаемый модуль Flash Player runtime. Напротив, популярные мобильные приложения Adobe Reader (Android/ios), eBookDroid (Android), iBooks (ios), к сожалению, не воспроизводят встроенное видео и, похоже, расширение функциональности в этом направлении не планируется.

Тем не менее, для этих платформ существуют приложения, которые, как заявлено разработчиками, воспроизводят встроенное видео: GoodReader (iOS), ezPDF (Android) и PDF Expert (iOS). Все три приложения являются платными. Для приложения ezPDF в Google Play существует пробная версия, что позволяет легко убедиться в его способности воспроизводить видео. По-видимому, для воспроизведения встроенных видео-объектов ezPDF использует инструменты, не связанные с технологией Adobe Flash, поскольку flash-приложения в ezPDF запустить не удастся, в том числе встраиваемые Strobe Media Playback и YouTube-плеер. Тем не менее, встроенные видео объекты ezPDF обнаруживает и успешно воспроизводит (правда, без учета настроек – flash-переменных), также поддерживаются действия дополнительных кнопок play/pause.

Подводя итог, к сожалению, следует признать, что отношения мобильных платформ с мультимедийными иллюстрациями складываются пока не лучшим образом.

Встраивание видео при помощи LaTeX

При верстке публикации в LaTeX в распоряжении автора имеется пакет `media9`, предназначенный для встраивания в PDF интерактивного содержимого [5]: Adobe Flash, 3D, файлов видео/аудио, а также потокового видео. Можно сказать, что `media9` по некоторым возможностям превосходит даже Adobe Acrobat; так, например, в Acrobat нет встроенных инструментов для добавления потокового видео с хостингов, в то время как в `media9` данная возможность предусмотрена изначально.

Для просмотра мультимедийных файлов в Adobe Reader всегда требуется Adobe Flash Player, который, начиная с версии 11, устанавливается как отдельный плагин. Для воспроизведения видео требуется дополнительное Flash-приложение, которое может быть встроено в документ или загружено "на лету". Для этого в пакет `media9` включена усовершенствованная версия flash-видеоплеера `StrobeMediaPlayback.swf`, рекомендованного Adobe, а также два простых плеера для видео и аудио – `VPlayer.swf` и `APlayer.swf`.

Для добавления мультимедиа любого типа предназначена команда `\includemedia`, а для добавления кнопок управления – команда `\mediabutton`. Формат команды `\includemedia` следующий:

```
\includemedia[<параметры и настройки>]
  {<замещающее изображение или текст>}
  {<Flash-приложение (SWF файл) или URL
    или 3D-объект в форматах PRC, U3D>}
```

Настройки мультимедийного содержимого представлены в виде пар ключ–значение, разделенных запятой. Полный список параметров подробно описан в [5]. Для видео-объекта необходимы следующие параметры.

addressource – каждый вызов данной команды встраивает очередной файл мультимедиа, требуемый для запуска основного Flash-приложения. Это имя файла видео, обложки медиа-плеера, дополнительные объекты 3D и т.п. Если уже встроенный файл используется в другой команде `\includemedia`, то этот параметр указывается снова, но файл заново не встраивается, а используется ссылка на первый объект, чтобы не увеличивать размер PDF-документа.

flashvars={<переменная_1=значение_1 & переменная_2=... >} – ActionScript переменные для настройки Flash-приложения: источник видео для медиаплеера (файл или потоковое видео), настройки громкости и т.п.

Аргумент **<замещающее изображение или текст>** определяет размер прямоугольной области на странице, которую занимает мультимедиа-объект. Замещающее изображение отображается, если мультимедийное содержимое не активировано или если оно отображается в отдельном окне.

Последний аргумент – главное интерактивное приложение, включаемое в PDF. Для Flash – это swf-файл или URL для YouTube видеоплеера. Локальный

swf становится частью документа, а URL требует подключения к интернету при открытии документа читателем.

Для добавления кнопок управления предназначена команда

`\mediabutton [<параметры>] {<надпись или изображение>}`

Запускаемые действия задаются через опции `mediacommand`, `3Dgotoview` (выбор вида 3D-объекта) и `jsaction` (JavaScript сценарий). Используя эти опции в различных комбинациях, кнопка может запускать множество различных действий. Все действия запускаются последовательно, но выполняются параллельно, не ожидая завершения предыдущих. Объект действия задается через метку, присвоенную медиа объекту в `includemedia`.

Опция `mediacommand` указывается в параметрах и имеет следующий формат:

`mediacommand = [метка : команда (список аргументов)]`

Команда с аргументами связывается с видео объектом при помощи метки. Например, команда

`mediacommand = physics_media:playPause`

запускает воспроизведение видео с меткой `physics_media`. А команда

**`mediacommand = physics_media:setSource
[(cubedensity.mp4)]`**

устанавливает источник видео из файла `cubedensity.mp4`, указанный в `\includemedia`.

Рассмотрим пример страницы в TeX, содержащей два видео. Первое воспроизводится при помощи `StrobeMediaPlayback.swf`, второе – при помощи `VPlayer.swf`. Второе видео содержит два источника (два файла видео), и для него создаются кнопки управления: кнопка `play/pause` и две кнопки выбора источника видео. Для второго видео обязательно наличие метки `label`, к которой привязываются действия кнопок. В обоих случаях видео запускается в окне, при этом на странице размещается постер с подписью.

```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{graphicx}
\usepackage{media9}
\begin{document}
\begin{figure}[]
\includemedia[
  activate=onclick,           % активация по щелчку на постере
  windowed = 1120x610,       % видео открывается в окне
  addresource=tempcontour.mp4, % подключаем файл видео
  flashvars={
    src=tempcontour.mp4
    &scaleMode=letterbox}]
{\includegraphics [width=\linewidth,keepaspectratio] {tempcontour}}
```

```

        {StrobeMediaPlayback.swf}      % плеер StrobeMediaPlayback
        \caption{"tempcontour"}      % подпись
\end{figure}
\begin{figure} [H]
\includemedia[
    label = physics_media,           % метка
    activate=onclick,                % активация по щелчку на постере
    windowed = 880x515,              % видео открывается в окне
    addresource=cubedensity.mp4,     % 1-й файл видео
    addresource=spheredensity.mp4,  % 2-й файл видео
    flashvars={
        source=cubedensity.mp4      % по-умолчанию - 1-й файл
        &hideBar=false
        &scaleMode=letterbox}]
    {\includegraphics[width=\linewidth,keepaspectratio]{density}}
    {VPlayer.swf}                   %кроссплатформенный плеер
VPlayer
    \caption{"cubedensity"}          % подпись
\mediabutton[
mediacommand=physics_media:playPause, % кнопка play/pause
overface=\color{blue}{\fbox{\strut Play/Pause}},
downface=\color{red}{\fbox{\strut Play/Pause}}
]{\fbox{\strut Play/Pause}}
\mediabutton[                       % кнопка выбора 1-го источника (файла)
mediacommand=physics_media:setSource [(cubedensity.mp4)]
]{\fbox{\strut Cube }}
\mediabutton[                       % кнопка выбора 2-го источника (файла)
mediacommand=physics_media:setSource [(spheredensity.mp4)]
]{\fbox{\strut Sphere }}
\end{figure}
\end{document}

```

Скомпилировав данный документ в LaTeX Live, получим PDF, содержащий два объекта видео на странице (рис. 7, 8). Видео на рис. 7 воспроизводится плеером StrobeMediaPlayer, имеющим полноценные элементы управления. Видео на рис. 8 воспроизводится кросс-платформенным плеером VPlayer, для которого можно переключать источники видео – кнопки Cube и Sphere.

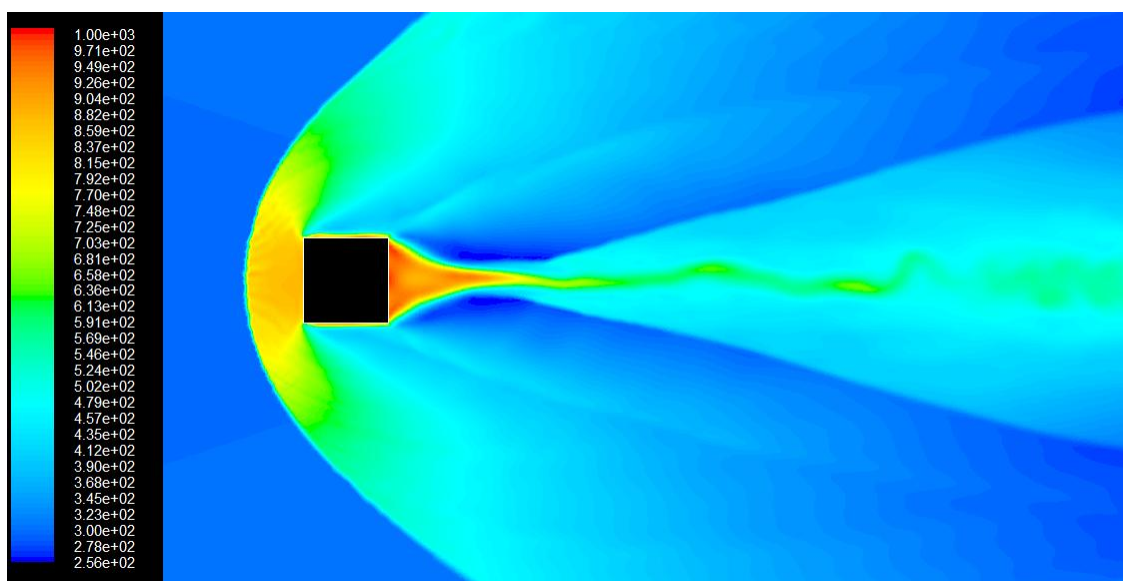


Рис. 7. Видео-объект, использующий StrobeMediaPlayer.swf

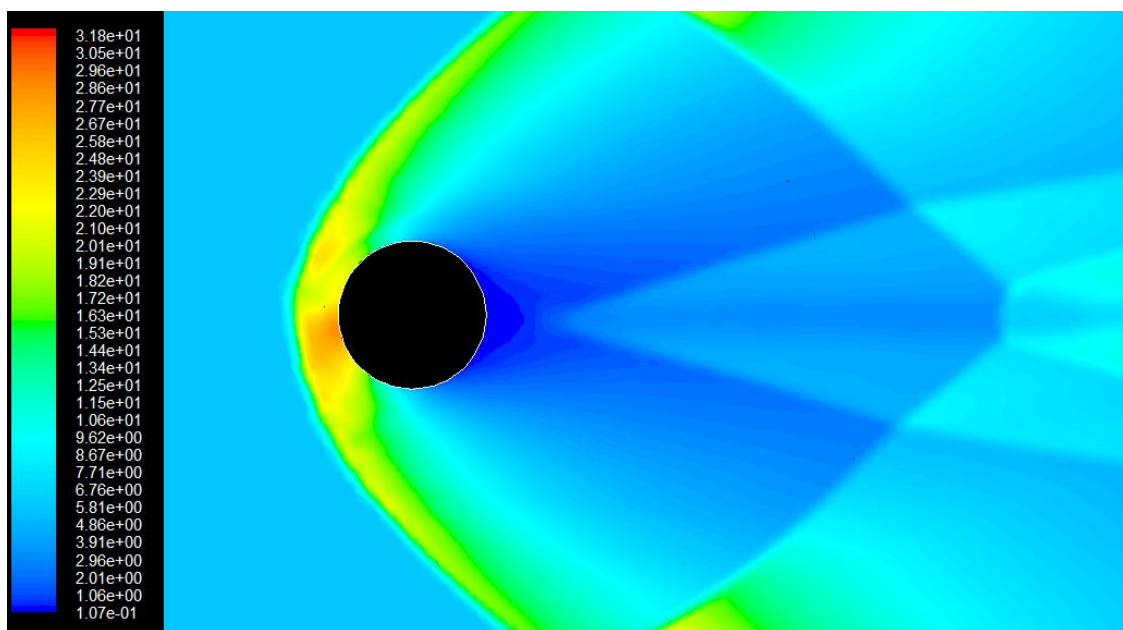


Рис. 8. Видео-объект, использующий VPlayer.swf.
Кнопки Cube и Sphere переключают источник видео

Стоит отметить, что при редактировании мультимедийных объектов в Adobe Acrobat часто возникают сложности, если эти объекты были встроены в PDF при помощи LaTeX. Так, объекты видео и 3D оказываются недоступны для функций JavaScript кода, специфицированных Acrobat [4]. Кроме того, мультимедийный объект и замещающее изображение существуют на странице по-отдельности, что усложняет их редактирование, но, будучи добавленными через Acrobat, они становятся единым объектом. Следует аккуратно относиться

к удалению и добавлению страниц в PDF-документе – всякий раз проверять работоспособность добавленных ранее мультимедийных объектов, скриптов и ссылок.

Встраивание потокового видео – YouTube

Пакет `media9` для TeX позволяет встраивать в документ потоковое видео с хостингов. На страницу помещается Flash-плагин (например, YouTube), для чего используется команда `\includemedia`. Последним аргументом передается ссылка на видеотреугольник, полученная на хостинге YouTube. Формат команды для включения потокового видео следующий:

```
\includemedia[настройки отображения, переменные flash]
  {постер или замещающий текст}
  {http://www.youtube.com/v/ID_видео?параметры}
```

После загрузки видео на YouTube необходимо выбрать постер (метка 1 на рис. 9) и в расширенных настройках разрешить встраивание видео (метка 2).

The image shows the YouTube video settings interface. The top part displays the video player and a sidebar with video information: Channel, Upload date (26 сентября 2014 г., 18:45), Duration (0:33), Original file (res.mp4), Views (1), Likes (0), Dislikes (0), and Comments (0). The URL is `http://youtu.be/-NLDPZRsE0g`. A red circle with the number '1' is placed over the video thumbnail selection area. Below the video player, the 'Advanced settings' tab is selected. Under 'Comments', the 'Allow users to rate this video' checkbox is checked. Under 'License and ownership', 'Standard YouTube License' is selected. Under 'Distribution', 'Everywhere' is selected. Under 'Reason for missing subtitles', 'Choose...' is selected. Under 'Embedding settings', the 'Allow embedding of this video' checkbox is checked and highlighted with a green circle and the number '2'. Other settings include 'Age restrictions' (not checked), 'Category' (Science & Technology), 'Video location' (Search), 'Video language' (Choose language), 'Recording date' (Today), '3D video' (No preference), and 'Video statistics' (Show statistics on the video page checked).

Рис. 9. Настройка видеоролика на YouTube: выбор постера (1), разрешение встраивания (2)

Ссылка для встраивания доступна на public-странице видеоролика – кнопка "Поделиться → HTML-код" (рис. 10). Выбрать функцию "Использовать старый код вставки". Для встраивания в \includemedia необходимо скопировать значение поля src.

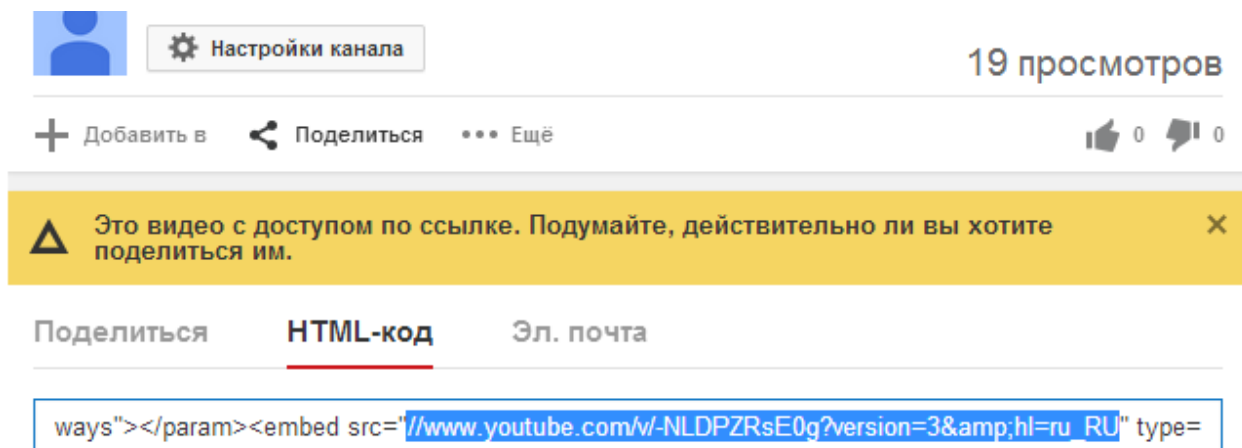


Рис. 10. Получение ссылки для встраивания видеоролика

В следующем примере в документ встраивается видео по ссылке <http://www.youtube.com/v/-NLDPZRsE0g>. Видео запускается по щелчку в отдельном окне с разрешением 608x540 точек, окно открывается в правом верхнем углу экрана. Flash-переменная `rel` запрещает или разрешает отображение связанных роликов после завершения данного видеоролика. Командой `\includegraphics` добавляем графический постер из файла `poster.jpg` с шириной, равной ширине строки, и с сохранением пропорций. Скомпилировав данный код, получим PDF-документ со встроенным потоковым видео, размещенным на YouTube (рис. 11).

```
\documentclass{article}
\usepackage[english]{babel}
\usepackage{graphicx}
\usepackage{media9}
\begin{document}
\begin{figure}[H]
\includemedia[
  activate=onclick,           % активация по щелчку
  windowed=608x540@tr,       % в окне в правом верхнем углу
  flashvars={
    modestbranding=1         % без логотипа YouTube
    &autohide=1               % авто-скрытие элементов управления
    &showinfo=0               % без заголовка перед загрузкой
    &rel=0                    % без связанных видеороликов после окончания
  }]{\includegraphics[width=\linewidth,keepaspectratio]{poster}}
{http://www.youtube.com/v/-NLDPZRsE0g?version=3&hl=ru_RU}
\caption{"CUDA Particles"}
\end{figure}
\end{document}
```

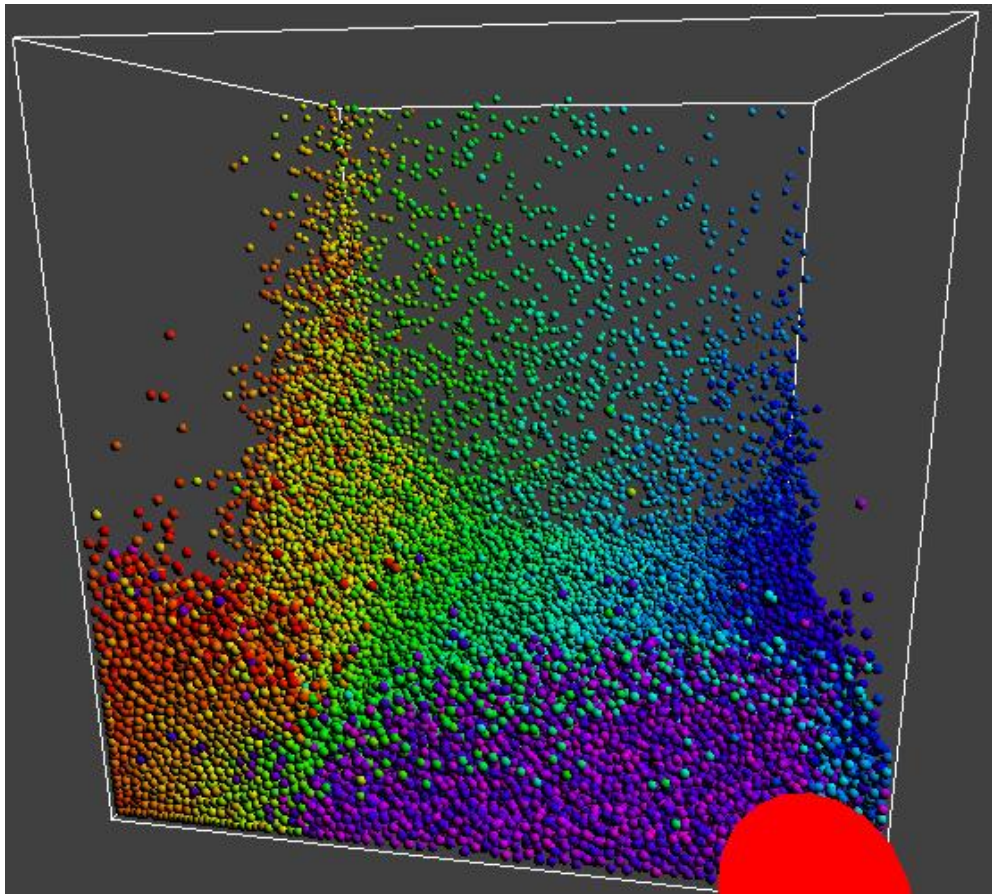


Рис. 11. Видео-объект, встроенный как потоковое видео с YouTube при помощи TeX с пакетом media9

Трехмерная иллюстрация

Наряду с видео полезным инструментом при создании научной публикации являются встраиваемые в PDF трехмерные графические объекты. Интерактивные трехмерные иллюстрации значительно улучшают восприятие информации читателем. Здесь он может самостоятельно настроить трехмерную сцену: расстояние до объекта, угловые координаты, отображение поверхностей и полигональной сетки, эффекты (освещение, прозрачность), масштаб и размер объекта. Появляется возможность представить как общий план, так и мельчайшие детали 3D-объекта в пределах одной интерактивной иллюстрации.

При необходимости автор может акцентировать внимание на определенных участках объекта, определив набор видов, которые читатель будет переключать в контексте одного 3D-объекта. Виды задаются небольшим набором параметров, практически не увеличивая размер документа, в отличие от аналогичного набора статических изображений.

Трехмерные графические объекты могут быть встроены в PDF-документ в форматах U3D (Universal 3D) и PRC (Product Representation Compact). Лучшую

компрессию полигональных сеток и более точное представление поверхностей обеспечивает формат PRC [6]. Тем не менее, формат U3D более распространен и доступен в виде Open Source библиотек. Многие среды моделирования и рендеринга предоставляют встроенные инструменты для сохранения трехмерных объектов в формате U3D. Для некоторых сред моделирования существуют подключаемые библиотеки, например, для Matlab рекомендуется открытая библиотека [fig2u3d](#) с дополнительным модулем [mesh2PDF](#).

Трехмерные объекты из среды Matlab сохраняются в формате U3D при помощи одноименной Matlab-функции `fig2u3d`, которая применяется после построения модели и сохранения ее в формате *.fig. Функция имеет следующий прототип:

FIG2U3D (*ax*, *fname*, *imgtype*, *addaxes*, *varargin*), где основные параметры:

- *ax* – объект осей координат, по умолчанию используется встроенный объект [gca](#);
- *fname* – имя файла с объектом *.fig;
- *imgtype* – графический формат для формирования замещающего изображения;
- *addaxes* – отображение осей координат в U3D файле, по умолчанию 0 – не отображать.

Сформировать PDF с интегрированным 3D-объектом, как и в случае с видео, можно при помощи LaTeX с пакетом `media9` или же, если документ создается иным способом, встроить непосредственно в PDF при помощи Adobe Acrobat.

В качестве примера построим 3D-модель в Matlab по следующему коду:

```
a=7;b=2;c=3;
u=linspace(0,2*pi,20);
v=linspace(0,2*pi,40);
[u,v]=meshgrid(u,v);
x=(a+b*cos(u)).*cos(v);
y=(a+b*cos(u)).*sin(v);
z=c*sin(u);

mhndl=mesh(x,y,z);
set(mhndl,...
'EdgeColor',[.6,.6,.6],...
'FaceAlpha',0.5,...
'EdgeAlpha',0.5);
axis equal

t=linspace(0,2*pi,200);
x=(a+b*cos(5*t)).*cos(2*t);
y=(a+b*cos(5*t)).*sin(2*t);
z=c*sin(5*t);
lhndl=line(x,y,z);
```

```
set(lhndl,...
'Color',[.625,0,0],...
'LineWidth',2)
```

```
view(135,30)
```

Построенную фигуру (рис. 12) сохраним в файл `thorus.fig`: File → Save as.... Созданные файлы помещаются в каталог проекта `C:\MF`. Теперь 3D-объект из файла `thorus.fig` сохраним в формате U3D, для чего в консоли Matlab выполним функцию: `fig2u3d(gca, 'thorus', '-png')`. Используем идентификатор координатных осей по умолчанию (`gca`), исходным файлом является `thorus.fig`, замещающее изображение запрашиваем в формате `png`.

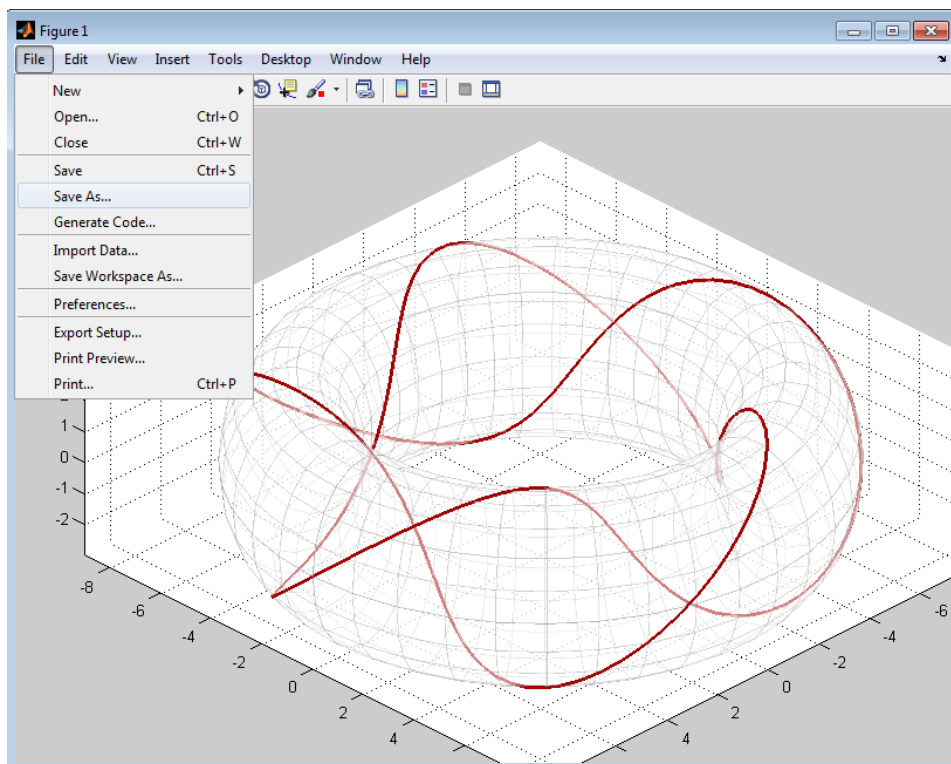


Рис. 12. Сохранение 3D-объекта в файл *.fig

Полученные файлы `C:\MF\thorus.u3d` и `C:\MF\thorus.png` будем использовать для встраивания 3D-объекта в PDF-документ.

При использовании LaTeX для верстки и встраивания 3D необходимо выполнить две операции:

- встроить 3D-объект в документ `.tex` и скомпилировать PDF;
- полученный PDF открыть в Adobe Reader, сформировать виды для объекта, скопировать параметры видов в документ `.vws`, подключить их к документу `.tex`, добавить кнопки и перекомпилировать.

В каталоге `C:\MF\` создадим документ `Doc3d.tex`. Используем пакет `media9`, команда `includemedia[параметры]{файл с 3D-объектом в u3d}`:

```
\documentclass{article}
```

```

\usepackage{graphicx}
\usepackage{media9}

\begin{document}
\begin{figure}
\centering
\includemedia[
width=\linewidth,height=\linewidth,
activate=pageopen,
3Dmenu
]{}{thorus.u3d}
\end{figure}
\end{document}

```

Скомпилируем в TexLive → pdfLaTeX, получим файл Doc3d.pdf. Откроем в AdobeReader, правой кнопкой мыши – 'контекстное меню → Generate Default View...'. 3D-объект будет масштабирован, так чтобы занимать все доступное пространство виджета, а в окне отладчика JavaScript будут показаны параметры объекта: поворот, масштаб, освещение и т.п. Эти параметры далее нам понадобятся для создания видов. Правой кнопкой мыши 'показать панель инструментов' – откроется меню с инструментами для манипулирования 3D-объектом. Включим прозрачный контур. Масштабируем и повернем объект для наилучшего отображения (рис. 13).

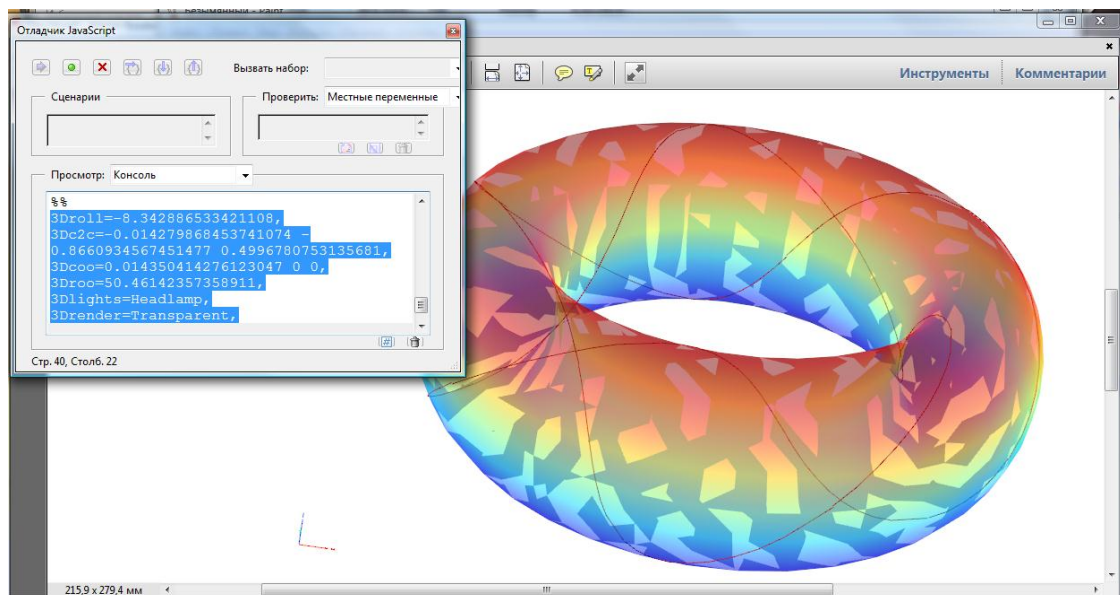


Рис. 13. Настройка стандартного вида для 3D-объекта

Создадим файл видов `thorus.vws`, имеющий следующий формат:

```

VIEW = Имя_Вида_1
3Droll = 'Начальный угол поворота камеры вокруг
          оптической оси (-1) C2C'
3Droo = 'расстояние от камеры до объекта'
3Dc2c = 'вектор начального положения камеры C2C'
...

```

```

END
VIEW = Имя_Вида_2
...

```

Правой кнопкой мыши – контекстное меню →GenerateDefaultView – скопируем параметры отображения и сохраним их в качестве основного вида VIEW = DEFAULT в файле thorus.vws. Таким же способом создадим еще два вида: TOP и FRONT, получится следующий файл видов:

```

VIEW = DEFAULT
3Droll = -11.191
3Droo = 50.461
3Dc2c = -0.789 0.0817 0.607
RENDERMODE = Transparent
END
VIEW = TOP
3Droll = -144.787
3Droo = 50.461
3Dc2c = -0.189 -0.118 0.974
END
VIEW = FRONT
3Droll = 0.014
3Droo = 50.461
3Dc2c = -0.891 -0.453 0.0019
END

```

Теперь подключим данный файл видов к документу *.tex, указав его в качестве параметра 3Dviews. Одновременно добавим некоторые параметры для медиа-объекта:

- метку 'label = thorus' для доступа к объекту при помощи скриптов и кнопок;
- свойство 'windowed = 800x600' для отображения в отдельном окне;
- замещающее изображение thorus.png;
- две кнопки, переключающие виды файла thorus.vws для 3D-объекта: "Следующий вид" и "Стандартный вид".

Документ Doc3D.tex примет следующий вид:

```

\documentclass{article}
\usepackage{graphicx}
\usepackage[utf8x]{inputenc} % поддержка UTF8
\usepackage[russian]{babel} % пакет поддержки русского языка
\usepackage{media9}

\begin{document}
\begin{figure}
\centering
\includemedia[
  label = thorus, % метка для доступа при помощи кнопок
  activate=onclick, % активация по клику
  windowed = 800x600@tr, % в окне, левый верхний угол

```

```

3Dmenu,
3Dlights=Headlamp,
3Drender=Transparent,
3Dviews=thorus.vws, % файл видов
% замещающее изображение
][\includegraphics[width=\linewidth,keepaspectratio]{thorus}}
{thorus.u3d} % файл U3D с 3D-объектом
% кнопка для переключения к следующему виду
\mediabutton[3Dgotoview=thorus:N]
{\fbox{Следующий вид}}
% кнопка для переключения к стандартному виду
\mediabutton[3Dgotoview=thorus:(DEFAULT)]
{\fbox{Стандартный вид}}
\end{figure}
\end{document}

```

Перекомпилировав документ Doc3d.tex в TexLive, получим окончательный PDF-документ со встроенным 3D-объектом. Кнопки "Следующий вид" и "Стандартный вид" используются для переключения между встроенными видами (рис. 14).

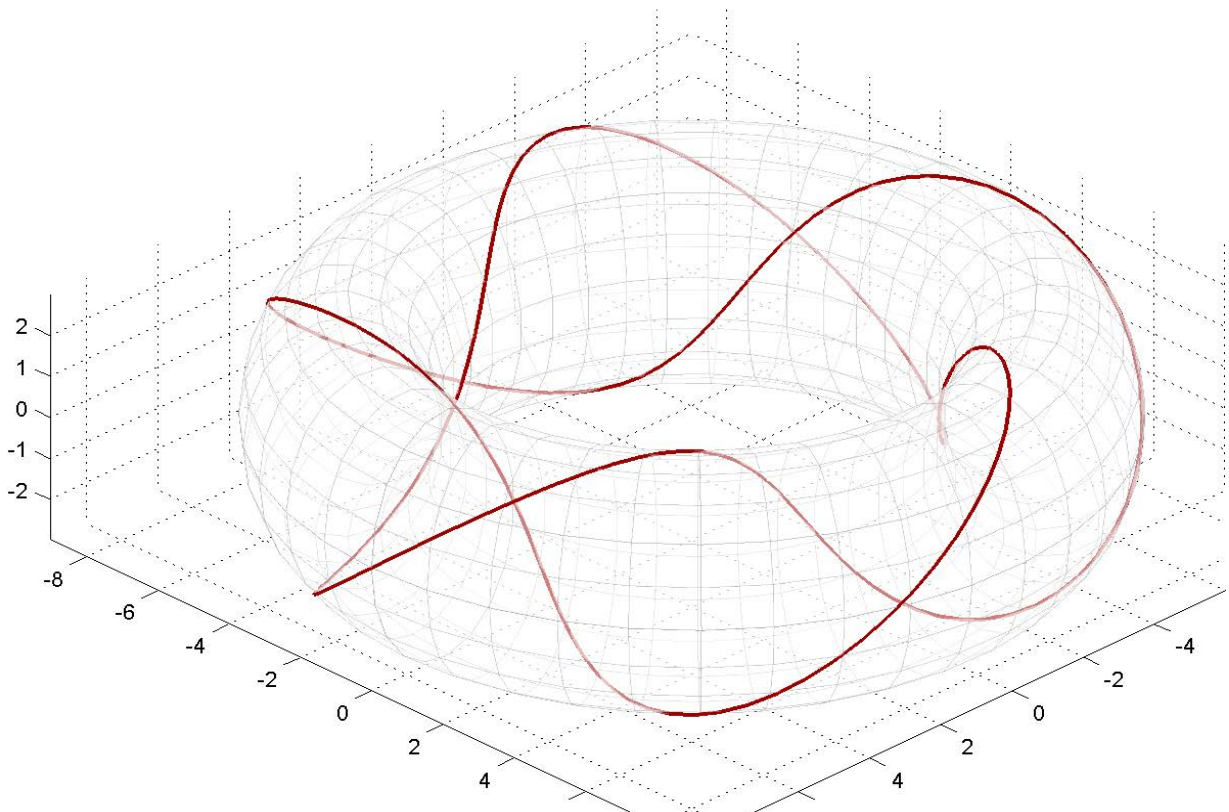


Рис. 14. 3D-иллюстрация в формате U3D, встроенная в PDF при помощи LaTeX Live

Для трехмерных иллюстраций, как и для видео, в документ рекомендуется встроить скрипт, проверяющий поддержку мультимедиа, в частности 3D-объектов:

```
c3d = this.getAnnots3D(pagenum) [0];
if(c3d == null){
    app.alert("Приложение не поддерживает встроенные
        3D-объекты. Для просмотра данного документа
        рекомендуется использовать Adobe Reader/Plugin", 3);
}
```

Подобный JavaScript-сценарий сообщит читателю о причине проблемы и подскажет подходящее средство просмотра. Для того чтобы сообщить читателю о наличии интерактивного 3D-объекта, на странице рекомендуется добавить кнопку "3D" (рис. 14), включающую/выключающую трехмерный объект в окне. К данной кнопке также стоит добавить скрипт с проверкой поддержки мультимедиа:

```
page = this.pageNum;
c3d = this.getAnnots3D(page) [1];
if(c3d){
    c3d.activated = !c3d.activated ;
}
else app.alert("Приложение не поддерживает 3D-объекты.
    Для просмотра данного документа рекомендуется
    использовать Adobe Reader/Plugin", 3);
```

Для встраивания 3D-объекта средствами Adobe Acrobat можно использовать инструмент 'Мультимедиа → 3D...' из меню 'Контент'. Последовательность шагов аналогична встраиванию видео объекта. Отметим, что во вкладке '3D' меню 'вставить 3D' можно настроить источники света, стили рендеринга и добавить виды объекта – встроенные или из файла. Добавить виды можно также при помощи контекстного меню 'Виды → управление видами' (рис. 15). Задав настройки очередному виду, его можно сохранить под определенным именем – TOP, MAIN, FRONT на рис. 15. Затем настроить переключение видов при помощи кнопок аналогично TeX-версии, для чего в свойстве кнопок задать действие 'Перейти в 3D-режим/... → Выбрать 3D-вид'.

Трехмерными объектами можно манипулировать при помощи JavaScript-сценариев. В [5] приводится пример 3D-модели, состоящей из нескольких объектов, которые воздействуют друг на друга, иллюстрируя принцип работы некоторого механизма в 3D. Такая возможность подчеркивает преимущества включения в документ трехмерных объектов, где в распоряжении автора оказывается достаточно богатый набор инструментов для создания мультимедийной интерактивной PDF-иллюстрации.

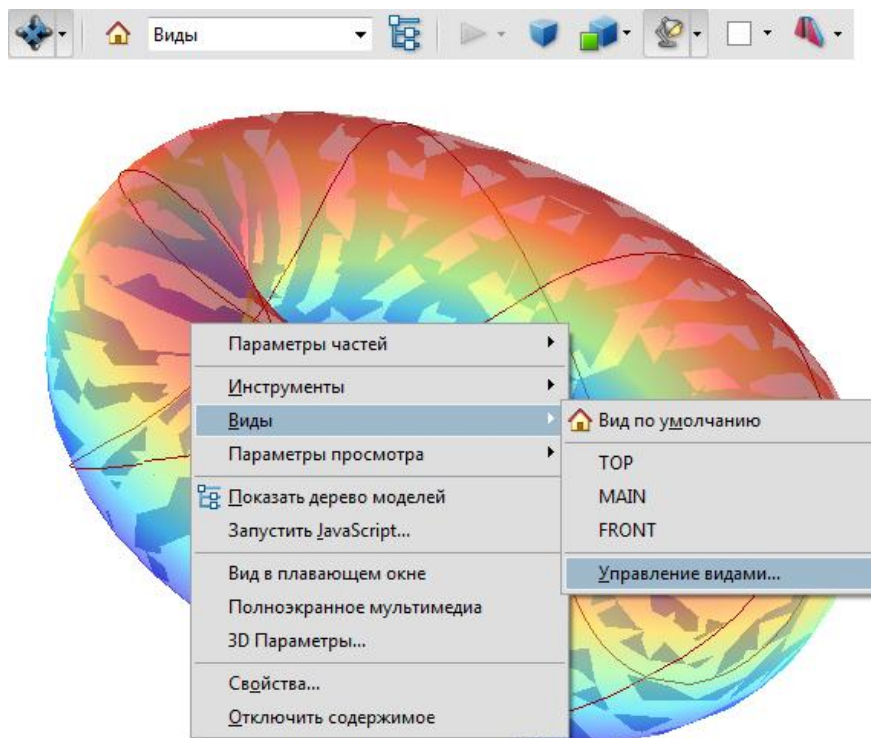


Рис. 15. Настройка видов в Adobe Acrobat

Заключение

Итак, показано, что размещение мультимедийных иллюстраций в формате PDF хотя и требует определенных навыков, но в целом не составляет особого труда. При сложившемся положении вещей автору научной публикации вполне по силам украсить ее мультимедиа. Один за другим выходят препринты ИПМ [7-9], глядя на которые удивляешься, как вообще раньше научные публикации обходились без анимации и трехмерных иллюстраций.

Хочется верить, что со временем работа с мультимедиа станет еще проще. С одной стороны, можно предположить, что браузеры, которые сейчас столь самоуверенно, безответственно и неумело берутся за визуализацию PDF-файла, все же осознают свою слабость и либо научатся полноценно, т.е. не хуже, чем Adobe, представлять PDF, либо скромно отойдут в сторонку, вернувшись к временам, когда в сфере визуализации PDF безраздельно господствовал Adobe Reader. С другой стороны, возможно, в сфере научных публикаций рано или поздно закончится эпоха безраздельного господства формата PDF, не вполне приспособленного для представления мультимедиа.

Сейчас позиции PDF как никогда крепки. Дело даже дошло до того, что наиболее авторитетный рейтинг сайтов научно-исследовательских организаций Webometrics [10] при подсчете баллов за количество научных публикаций учитывает размещенные на сайте организации файлы формата PDF, но не HTML!

Вместе с тем мир HTML много богаче мира PDF, как в части организации связей в области семантического веба, так и, главное, в части динамических

экранных конструкций. Продолжая ориентироваться на PDF, научные публикации заметно обкрадывают себя, пренебрегая широчайшими возможностями HTML. Если лет десять назад обращение к PDF можно было еще оправдать, например, неудобством представления в HTML математических формул, то сейчас, с появлением мощного инструментария MathJax [11], преград на пути научной публикации в HTML практически не осталось.

Так или иначе, мультимедийные иллюстрации в онлайн-научной публикации вполне реализуемы уже сегодня и станут еще доступнее в ближайшем будущем.

Литература

- [1] Movies on OS X. – <http://www.mathworks.com/matlabcentral/answers/141591-movies-on-os-x>
- [2] Matlab Documentation. Import and export audio and video files – <http://www.mathworks.com/help/matlab/reading-and-writing-files.html>.
- [3] FFmpeg – a complete, cross-platform solution to record, convert and stream audio and video – <https://www.ffmpeg.org/>
- [4] Adobe Systems Inc., JavaScript for Acrobat API Reference – Adobe Acrobat SDK Version 8.1, 2007
- [5] Alexander Grahn – The media9 Package – <http://ctan.org/pkg/media9>
- [6] PRC Format Specification – http://help.adobe.com/livedocs/acrobat_sdk/9/Acrobat9_HTMLHelp/API_References/PRCReference/PRC_Format_Specification/index.html
- [7] Малинецкий Г.Г., Фаллер Д.С. Сценарии перехода к хаосу в двухмодовой системе для моделей «реакция-диффузия» // Препринты ИПМ им. М.В.Келдыша. 2013. № 67. 36 с. URL: <http://library.keldysh.ru/preprint.asp?id=2013-67>
- [8] Ovchinnikov M.Yu., Penkov V.I., Malphrus Benjamin, Brown Kevin, Roldugin D.S. Active magnetic attitude control algorithms for a Cubesat for astrophysics research // Keldysh Institute Preprints. 2014. No. 47. 18 p. URL: <http://library.keldysh.ru/preprint.asp?id=2014-47&lg=e>
- [9] Шестаков С.А., Иванов Д.С. Оптимизационные задачи при управлении групповым полетом спутников с помощью переброса массы // Препринты ИПМ им. М.В.Келдыша. 2014. № 69. 24 с. URL: <http://library.keldysh.ru/preprint.asp?id=2014-69>

- [10] Ranking web of research centers – <http://research.webometrics.info>
- [11] MathJax – an open source JavaScript display engine for mathematics that works in all browsers – <http://www.mathjax.org/>