

Связка OpenOffice Calc + Basic для инженерных расчетов

Myooo.ru

Введение

Производительность труда современного инженера зависит от эффективности использования им персонального компьютера. Навыки программирования — один из ключевых факторов эффективной работы инженера в большинстве приложений, от AutoCAD до офисного пакета.

В статье речь пойдет о программировании расчетов, выполняемых по инженерным методикам, по алгоритмам ГОСТ, СНиП, СП, Eurocode и других аналогичных документов. Статья, в основном, будет касаться выполнения несложных инженерных расчетов строительных конструкций, однако все изложенное ниже применимо и к расчетам в других сферах деятельности: машиностроении, промышленной безопасности, экономике и финансах.

Допустим, нужно выполнить сбор нагрузок, рассчитать прочность балки или узлового соединения элементов. Есть несколько способов для реализации задачи:

I. Все расчеты выполнить вручную

Взять чистый лист с рамкой, карандаш с ластиком (или авторучку с корректором), калькулятор, нормативный документ и сделать то, что нужно.

Достоинства

1. Не нужно тратить деньги на компьютер, операционную систему и программы.
2. При наличии соответствующих навыков на выходе получается своего рода произведение искусства — аккуратный почерк, приятные глазу схемы и рисунки. Пометки и правки легко расположить таким образом, что текст останется удобочитаемым.
3. Некоторые заказчики легко соглашаются заплатить за работу максимальную цену, если узнают, что она связана с затратами «ручного» труда.

Недостатки

1. Сложность внесения изменений. Расчет сделан, подшит, внезапно меняется техническое задание. В дело идут корректор, ластик, ножницы, полоски бумаги — вся «красота» улетучивается. Очень часто приходится переделывать все заново. Вдобавок чертежник, работающий совместно с расчетчиком, должен будет некоторое время ждать новых результатов, чтобы внести изменения в проект.
2. Необходимость тщательной проверки. Каждому новому расчету нужна индивидуальная проверка, включая проверку элементарных вычислений. Увы, невнимательным может оказаться любой человек.

II. Купить специальную программу или набор программ и выполнять расчеты с ее помощью

Достоинства

1. Не во всякой организации работают инженеры, способные разобраться с общей нормативной методикой расчета. Используя готовую программу, можно до некоторой степени рассчитывать на упрощение задачи, поскольку расчет там сводится только к вводу необходимых данных.
2. Легче переделывать расчеты, легко отслеживать изменения результатов при вариации

исходных данных.

Недостатки

1. Стоимость. Программы стоят денег. Полный набор программ для расчета одних только металлоконструкций потянет на несколько месячных зарплат среднего провинциального инженера. Доходы организации могут упасть, а цены на софт — вряд ли
2. Старение. Нормативы меняются, программы устаревают, некоторые из них отказываются полноценно работать на новых версиях и сервис-паках операционных систем. Неудобство доставляет и необходимость использования электронного ключа защиты, который нужно носить с собой.
3. Ошибки. К сожалению, любая программа может ошибаться, в том числе и платная. Разработчики программ или органы сертификации не несут ответственности даже в том случае, если ошибка в программе станет причиной крупной аварии, и это будет доказано при расследовании причин. Ошибки часто возникают от неверного взаимодействия с программой, неправильной интерпретации пользователем ее возможностей. Даже владея методикой, бывает не так просто разобраться, какой смысл вложен в понятия и величины, которыми разработчики предлагают иногда оперировать в погоне за простотой и универсальностью. Программы на продажу редко пишутся практикующими инженерами, а разработчики программ часто упрощают отдельные фрагменты алгоритма или не предусматривают ввод данных, которые могут оказать влияние на результат.
4. Несовершенство отчета. Отчет программы может оказаться как излишне подробным, содержащим ненужные мелочи, так и неоправданно сокращенным. Промежуточные выкладки часто не приводятся, что затрудняет контроль правильности расчета.

III. Скачать бесплатную программу или шаблон

Достоинства

Совпадают с предыдущим способом, только не нужно покупать, тратя деньги, которые никогда не бывают лишними. Кроме того, тот или иной расчет бывает нужен эпизодически.

Недостатки

Во-первых, по-настоящему хороших бесплатных программ не так много. Во-вторых, такого рода шаблоны и программы нужно тестировать и проверять более придирчиво, чем платные. Вероятность ошибок в них намного больше, чем в коммерческих программах. Если программа закрытая, то в случае возникновения ошибки вы не сможете понять ее причины и исправить самостоятельно.

IV. Написать собственную программу и выполнять расчеты с ее помощью

Поскольку вся статья и посвящена этому способу, то и достоинства с недостатками будут обсуждаться далее по тексту. Сразу можно отметить, что в этом случае не нужно тратить деньги на покупку готовых программ и время на поиск бесплатных. Расчетные приложения, как правило, простые, и их разработка требует в большей мере знания самого предмета, нежели программирования. В процессе разработки вы будете иметь возможность протестировать программу самостоятельно, а при возникновении ошибок — отследить причины и исправить их самостоятельно.

Все описанные выше способы осуществления расчетов подтвердили свою жизнеспособность на практике и каждый, благодаря своим специфическим достоинствам имеет право на существование. Так как любой выполненный расчет, независимо от способа

нуждается в проверке, то можно варьировать инструменты, используя один способ для проверки результатов другого.

В чем писать расчеты?

Существует большое количество языков и сред программирования. Выбор языка — не предмет обсуждения настоящей статьи; в основном, каждый пишет в том, в чем умеет.

Какие приложения предпочтительней — самостоятельные, которые могут быть скомпилированы в исполняемый файл, или в виде расчетных шаблонов с использованием интерпретируемых языков?

Основное преимущество последних — простота, скорость разработки и тестирования. Недостаток — кроме операционной системы нужен соответствующий пакет. Само по себе наличие пакета-«посредника» не является недостатком (например, для работы .exe файлов в Windows также необходимы скрытые от глаз пользователя библиотеки или соответствующая версия .NET Framework). Гораздо более серьезным осложнением является то, что большинство пакетов такого рода, во-первых, имеют чрезмерный для инженерной деятельности математический или оформительский уклон, во-вторых, стоят приличных денег.

Одним из оптимальных по простоте и функциональности решений является MS Excel в составе пакета MS Office. Электронные таблицы используются для инженерных расчетов достаточно широко, на них написано много удобных и полезных инженерных шаблонов. Но хорошо проработанные экселевские шаблоны чаще всего не бесплатны: за рубежом цена их за штуку в среднем варьируется от \$30 до \$200. Легальная установка самого MS Office (не считая обучения) также требует материальных затрат, довольно существенных, особенно если покупать коробочные версии на существующие машины.

Другой офисный пакет — OpenOffice.org — этого недостатка лишен, а все достоинства в части выполнения с его помощью расчетов присутствуют в полном объеме.

Многие инженеры, использующие электронные таблицы, настороженно относятся к VBA в Excel и, соответственно, к OpenOffice.org Basic в Calc. Негласные «правила хорошего тона» предполагают, что расчетный шаблон должен быть, по возможности, создан базовыми средствами программы без пользовательских функций и макросов. Сложно сказать, в чем причины такого предубеждения. Я полагаю, что, скорее всего, дело в пугающем стандартном предупреждении о возможных макровирусах, которое появляется при настройке на «средний уровень безопасности» и на подсознательном уровне не располагает связываться с макросами вообще. В результате для того, чтобы запрограммировать более или менее сложный расчет, инженеры вынуждены прибегать к ухищрениям вроде многоуровневого пересчета по формулам в скрытых ячейках. В таких случаях контроль за алгоритмом затруднен, часто приходится действовать на ощупь. Не говоря уже о трудночитаемом «коде» вида « $\$A\$1/(\$B2*IF(IF(.....$ », ввести и проверить который бывает не так просто, а случайно повредить при правке легко.

После многочасовых бдений за монитором и нескольких неожиданных «фортелей» с результатами я отказался от такого способа сочинения «правильных» шаблонов и окончательно остановил свой выбор на использовании макроязыка. Программа на мой взгляд должна не просто работать, а работать понятно, отлаживаться легко, а любая неточность в коде — мгновенно выхватываться и исправляться.

Замечу также, что в тех случаях, когда большая часть расчета реализована и отлажена средствами языка OpenOffice.org Basic, не составляет труда «перевести» программу в отдельно работающее «оконное» приложение при помощи любого подходящего компилятора Basic, например, кроссплатформенного [IDE RapidQ Basic](#) или [FreeBasic](#).

После того, как вопрос расчетной оценки той или иной величины будет вами решен, у вас появится возможность улучшать проектные параметры конструкций и узлов, например, с помощью [настройки](#) нелинейной оптимизации.

Как написать расчетный шаблон, с чего начать?

Для начала достаточно запустить OpenOffice.org Calc, сохранить файл со своим именем, как .ods файл и нажать на клавиатуре [Alt+F11]. Появляется окно **OpenOffice.org Basic Macros** (рис. 1).

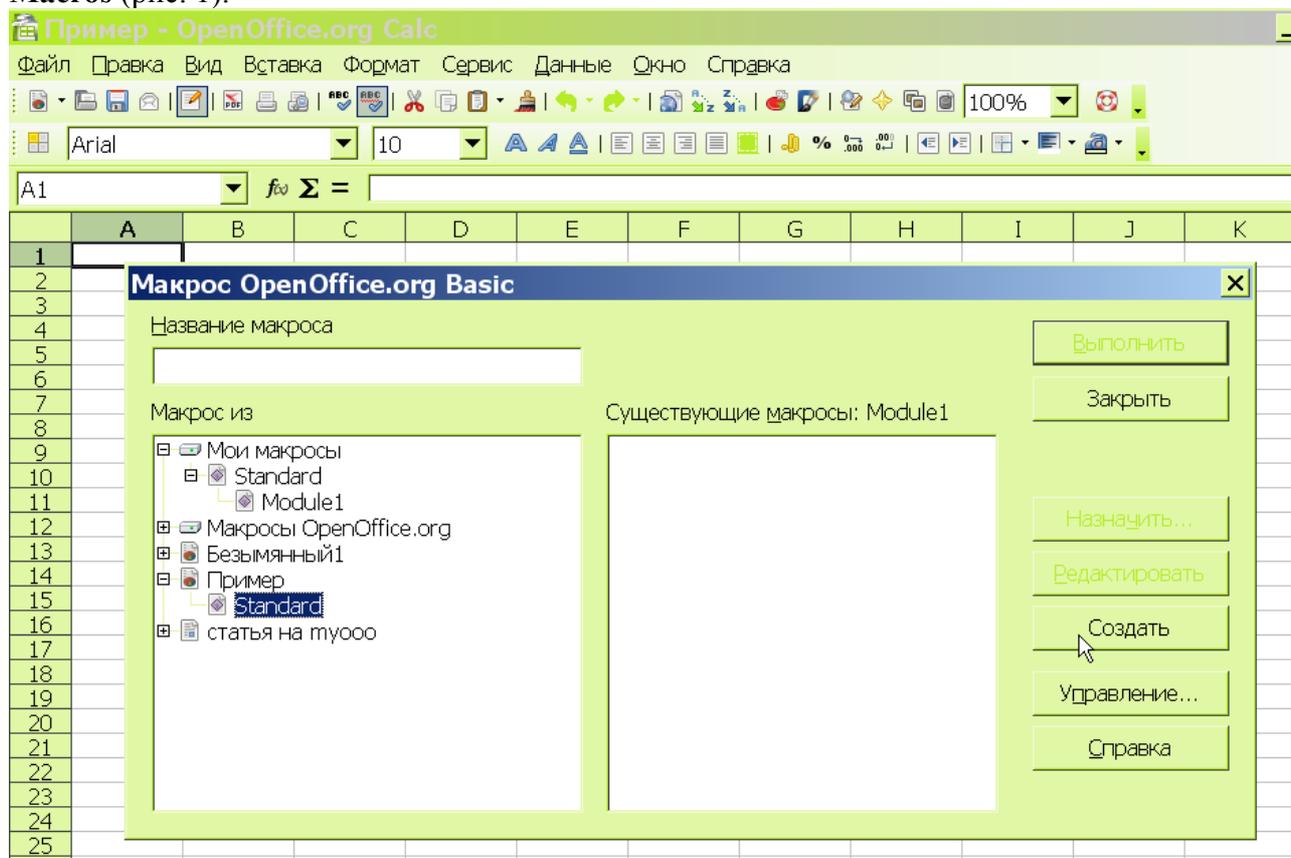


Рис. 1. Окно выбора макроса

Находим в окне **Макрос из (Macro From...)** свой файл и жмем **Создать (New)**. Новый модуль, в котором будет содержаться требуемый код, по умолчанию — **Module1**, создан, и на мониторе появляется активное окно редактора кода, где собственно и будет работать программа (рис. 2).

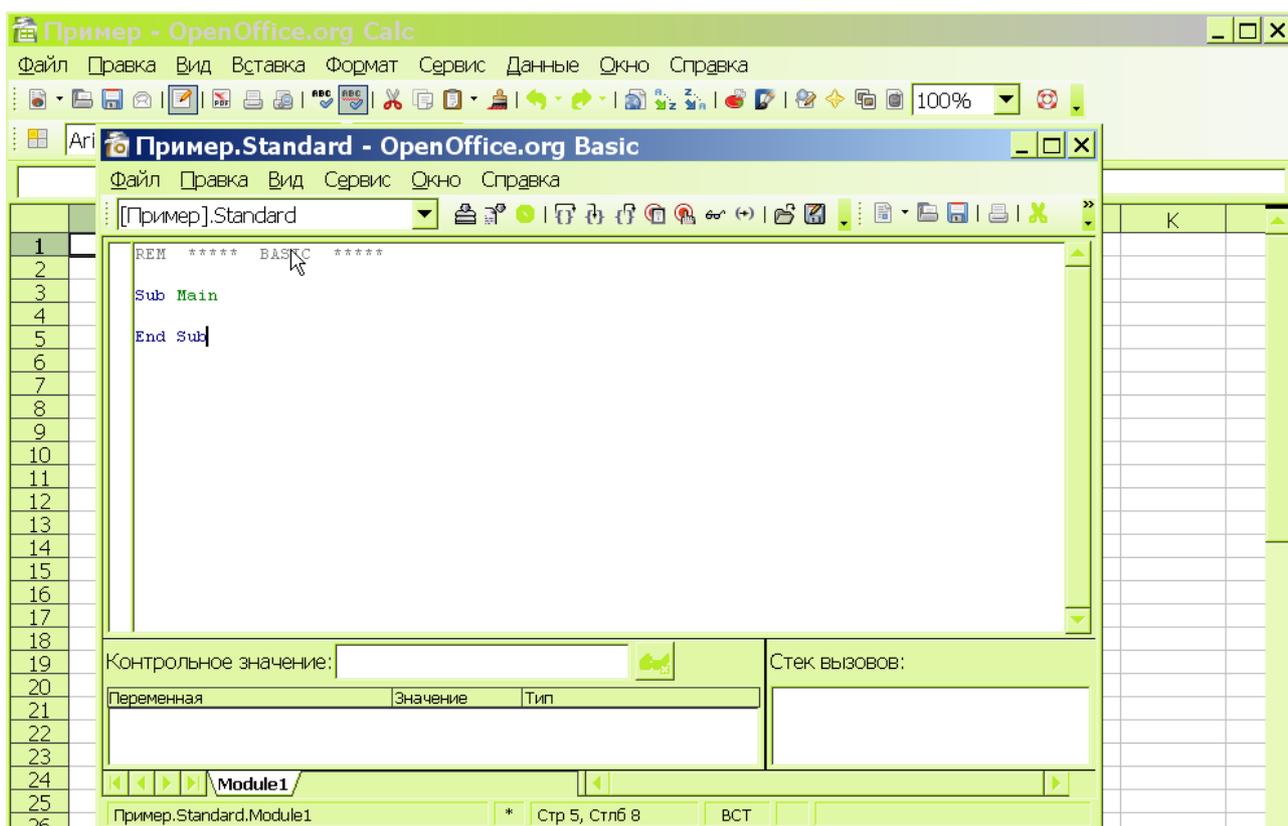


Рис. 2. Окно редактора кода

Объяснять по шагам, как начать программировать, я не буду, замечу только что доступной литературы по OpenOffice Basic достаточно для пользователя любого уровня. Более того, для начинающего сойдет литература и по VBA для Excel, особенно в части написания пользовательских функций (в свое время я начинал осваивать этот инструмент по экселевской «Библии»).

Вот несколько видеуроков, иллюстрирующих написание и работу простейших функций (для того, чтобы разглядеть детали, разверните видео на весь экран).

[Первый ролик](#) — простейшая функция в одну строчку, не считая стандартного объявления функции, завершения и комментария. Переменные не объявляются, подразумевается тип Variant. Известно, что такой подход в ряде случаев может приводить к ошибкам интерпретации значения, но раз существует «универсальная» переменная, то где-то она должна быть использована?

Этот пример — не очень удачный в том смысле, что здесь можно было запросто обойтись и стандартной формулой:

$$=2*B1*B4^3/12+2*B1*B4*(B2/2-B4/2)^2+B3*(B2-2*B4)^3/12$$

Более того, в данном случае [возможен вариант](#), при котором можно было вбить формулу $=2*b*t^3/12+2*b*t*(h/2-t/2)^2+s*(h-2*t)^3/12$ непосредственно в ячейку, предложив программе самой распознать переменные и присвоить им соответствующие значения. Я не использую его никогда в своей практике — не люблю, когда программа "думает" за меня. Но все способы, тем не менее, работают, что и требовалось доказать.

Рассмотрим [другой пример](#). Тут требуется найти решение кубического уравнения, к которому сводится задача расчета оттяжки или кабеля, провисающего под поперечной нагрузкой. Само решение легко находится в [wxMaxima](#). Выбираем нужный (вещественный) корень уравнения, выбираем в контекстном меню «скопировать как текст», копируем и вставляем в код функции. Маленькая правка (замена "sqrt" на бэйсиковское "sqr") — и функция готова. Меньше ручных операций — меньше вероятность ошибки (хотя проверка, в любом случае, не помешает никогда).

Наконец, [рассмотрим пример](#), когда функция возвращает не число, а целый отчет с текстом и численными значениями. См. код:

```

Function BeamCalculation(q,L,Jx,Wx,Em)
'Возвращает результаты расчета балки в виде текстовой строки
'Переменные
'q - нагрузка в кгс/м
'L - пролет балки, м
'Jx - момент инерции сечения, см4
'Wx - момент сопротивления сечения, см3
'Em - модуль Юнга материала (кгс/см2)

'Изгибающий момент, кгс*м
Mom=q*L^2/8

'Напряжение, кгс/см2
sigma=Mom*100/Wx

'Прогиб балки, см
ybeam=(5/384)*(q/100)*(L*100)^4/(Em*Jx)

'Результат функции
BeamCalculation="Изгибающий момент M= " & Str(Round(Mom,2)) & " кгс*м; Максимальное напряжение - "&
Str(Round(sigma,0)) & " кгс/см2; Прогиб балки - " & Str(Round(ybeam,1)) & " см, составляет 1/"&
Str(Round(L*100/ybeam,0)) & " от пролета."

End Function

Function Round(xxx,nnn)
'Пользовательская функция возвращает округленное до nnn знаков число aaa
Round=Fix(xxx * 10^nnn + 0.5) / 10^nnn
End Function

```

Функция округления Round(x,a) в VBA есть, а в OpenOffice.org Basic — отсутствует. Достаточно [воспользоваться](#) предложением Эндрю Питоньяка, чтобы создать свою.

Есть и другие разновидности функций в OpenOffice.org Basic. Например, результатом функции может быть не только число или текстовая переменная, но и произвольный массив чисел и строк. Такие функции вводятся в ячейку с одновременным нажатием клавиш "Ctrl+Shift+Enter" и размещают результаты на заданной области рабочей книги. [См. пример:](#)

```

Function BeamCalculation2(array)
'Возвращает результаты расчета балки в виде смешанного массива
'(вводится через Ctrl+Shift+Enter)
'Переменные получаем из исходного вектор-столбца array
'q - нагрузка в кгс/м
q=array(1,1)
'L - пролет балки, м
L=array(2,1)
'Jx - момент инерции сечения, см4
Jx=array(3,1)
'Wx - момент сопротивления сечения, см3
Wx=array(4,1)
'Em - модуль Юнга материала (кгс/см2)
Em=array(5,1)

'Изгибающий момент, кгс*м
Mom=q*L^2/8

'Напряжение, кгс/см2
sigma=Mom*100/Wx

'Прогиб балки, см
ybeam=(5/384)*(q/100)*(L*100)^4/(Em*Jx)

'Создаем массив 3 x 3 для результатов
Dim res(1 To 3,1 To 3)

res(1,1)="Изгибающий момент M = "
res(1,2)=Mom
res(1,3)=" кгс*м"

res(2,1)="Напряжение = "
res(2,2)=sigma
res(2,3)=" кгс/см2"

res(3,1)="Прогиб балки = "
res(3,2)=ybeam
res(3,3)=" см"

BeamCalculation2=res

End Function

```

В целом, для написания пользовательских функций для инженерных расчетов никаких особых знаний и навыков программирования не требуется. Все, чем я обходился до сих пор в

95% случаев, это:

- оператор If...Then...Else,
- инструкция Select...Case,
- циклы For...Next и Do ... Loop,
- действия с массивами и элементами массивов.

Допустим, что вы уже освоили простые функции на OpenOffice.org Basic, самостоятельно написали несколько собственных примеров и убедились в их работоспособности.

Рабочий шаблон начинается с освоения расчетной методики. Желательно иметь перед глазами рассчитанный вами вручную пример, чтобы по ходу отладки у вас было с чем сравнивать. Вот приблизительный перечень вопросов, которые вам нужно решить:

1. Для какой части расчета достаточно стандартных формул электронных таблиц, а для какой целесообразней воспользоваться OpenOffice.org Basic?
2. Каков примерный перечень пользовательских функций для расчета и возможность использования тех функций, которые у вас уже есть в наличии? Нерационально помещать весь расчет в одну или две функции, лучше всего предусмотреть отдельные функции для каждой проверки и для каждого коэффициента. Каждая из таких функций отлаживается и проверяется отдельно.
3. Сколько будет переменных, каковы их имена (и, возможно, типы, отличные от Variant по умолчанию), как исходных для пользовательских функций, так и промежуточных в теле функции?

Для начала лучше ограничиться самым простым внешним оформлением шаблона, потому что оформительскую часть работы смешивать с частью, ответственной за конечные результаты крайне нежелательно.

Можно построить блок-схему для расчета, но это совсем не обязательно: если расчет несложен, и вы уже делали его ранее, то блок-схема должна быть у вас в голове.

Оформление расчетного шаблона

Все написанное мной ниже — мое субъективное мнение, потому что в таком деле, как дизайн чего-либо, двух одинаковых взглядов быть не может. Много зависит от того, за какое время мы хотим получить требуемый результат. Я, чаще всего, исхожу из принципа минимализма, предпочитая потратить больше времени на тестирование самого расчета и продумывание дополнительных опций, делающих произведение более гибким в использовании, чем собственно на оформление.

Самый незатейливый способ — делать шаблон «в строчку», наподобие изображенного на рисунке 3. Пояснения есть, исходные данные развернуты, приведены результаты промежуточные и окончательные. В принципе, всего достаточно.

Весьма желательно отделять часть листа с исходными данными от результатов, по тому же принципу, по которому оформлялись школьные задачи по физике ("дано", "решение"). Подход к оформлению, при котором изменяемые и неизменяемые ячейки оказываются беспорядочно разбросанными по всему листу и маркируются определенными цветами, на мой взгляд, менее удачен.

В случае, когда вводимые данные - детерминированные (например — обозначение профиля из сортамента, расчетный случай и т.д.) лучше всего оформлять соответствующие ячейки в виде выпадающего списка (элементы вводятся в меню "Данные" > "Проверка..." > "Критерий" / "Допустимый тип данных" > "список").

Ячейки с формулами целесообразно защищать от случайных изменений, даже в тех случаях, когда шаблон предназначается для личного использования, без передачи на сторону.

	A	B	C	D	E	F	G	H
1	РАСЧЕТ СТОЙКИ ФАХВЕРКА							
2								
3	Исходные данные:				Сечение			
4	Момент инерции профиля $J_x =$	1405	см ⁴		160x6			
5	Момент сопротивления профиля $W_x =$	175.6	см ³		160x5.5			
6					160x6			
7	Норм. ветровое давление, $W_0 =$	38	кгс/м ²		160x6.5			
8	Предельная частота $f_z =$	3.8	Гц		160x7			
9	Нач. уровень отсчета высоты сооружения, $z_0 =$	0	м		160x7.5			
10	Тип местности	B			160x8			
11	Высота (пролет) стойки, $H =$	6.129	м		180x5			
12	Ширина грузовой полосы (шаг стоек)	6	м		180x5.5			
13	Аэродинамич. коэффициент, $c =$	1			180x6			
14	Масса стенового заполнения	55	кг/м ²		180x6.5			
15					180x7			
16	Результаты расчета:							
17	Кoeff. по табл. 6 СНиП 2.01.07-85* $k =$	0.534						
18	Кoeff. по табл. 7 СНиП 2.01.07-85* $\zeta =$	1.184						
19	Кoeff. корреляции пульсаций	0.88						
20	Скорость ветра, $v_{\text{расч}} =$	121.7	м/с					

Рис. 3. Пример оформления шаблона

Как быть с формулами? Как проверяющий поймет, что вы насчитали? Можно ограничиться поясняющей надписью со ссылкой на источник. Например «коэффициент такой то, по формуле ... СНиП...». Можно набрать методику в OpenOffice.org Writer в отдельном файле, который сохраняется и распечатывается вместе с данным расчетом (можно скопировать методику из [«Стройконсультанта»](#)). Формулы с отображением «этажей» можно не только набирать в OpenOffice.org Math, но и, например, копировать из wxMaxima, как изображение.

Если расчет основан на функциях OpenOffice.org Basic, то можно даже распечатать и приложить к расчету хорошо структурированный и закомментированный код.

А может быть и [такое многосложное решение](#) со вставкой формул Math с помощью кода.

Отмечу только то, что какой бы программой вы не пользовались и как бы ни было продумано оформление, на практике редко удастся избежать ручных пояснений и правок на твердой копии, после распечатки отчета. Если, конечно, все что вы насчитали, не является для вас непонятной абстракцией ;-).

Отладка расчетов в электронных таблицах

Для проверки отдельных значений вы можете воспользоваться результатами ручного расчета или результатами, полученными в другой программе (если уверены, что она возвращает именно то, что вы имеете в виду).

Один из наиболее удобных способов отладки функций, возвращающих численное значение — построение диаграмм. Пользуясь графиком, удобно отслеживать зависимости между переменными и результатами, соотносясь с их физическим смыслом. Расчетные методики довольно часто включают разные аналитические выражения для одной и той же величины. Например, коэффициент продольного изгиба для расчета на устойчивость вычисляется по разным формулам для гибких стержней, выпучивающихся до достижения напряжениями предела пропорциональности и после него, для расчета изгибаемых железобетонных элементов по нормальным сечениям рассматриваются варианты для двух случаев — пластичного и хрупкого разрушения, в зависимости от относительной высоты сжатой зоны. На соответствующих графиках точки перехода между разными расчетными случаями могут иметь скачок в значении функции или перелом (скачок в производной).

Для функций, заданных таблично, нужно убедиться в плавности соответствующих кривых (которая обеспечивается при правильном вводе табличных значений в код программы).

Некоторые частные вопросы

Размерности физических величин

На практике размерности физических величин могут отличаться от единиц СИ. Так повелось, что площадь сечений прокатных профилей удобней указывать в квадратных сантиметрах, в качестве единиц силы многие используют по старинке тонн-силы, для напряжений, вместо паскалей и килопаскалей, — килограммы на квадратный сантиметр, длины и катеты сварных швов измеряют в миллиметрах и так далее. OpenOffice.org Basic и OpenOffice.org Calc никаких специальных настроек для размерностей не содержит, поэтому, создавая расчетный шаблон, вы сами должны понимать, в чем измеряются те или иные переменные в ваших расчетах.

На практике, для того чтобы формулы не содержали в себе «переводных множителей» удобно с самого начала привести все величины к одним единицам. Вводить новую переменную при этом необязательно, поскольку в начале кода она может измеряться в одном, далее — в другом. Пусть, например, переменная Wx означает момент сопротивления в кубических сантиметрах, а по ходу расчета у вас участвует еще и длина в метрах. Тогда для удобства целесообразно пересчитать:

```
' Перевод из см3 в м3  
Wx=Wx*1e-6
```

Окончательный результат — напряжения в кгс/см² переводите из кгс/м².

```
' Перевод напряжений из кгс/м2 в кгс/см2  
sigma=sigma*1e-4
```

Скорость исполнения кода функций

Практически проблема с медленным счетом возникает нечасто, одна-две секунды на обновление всего расчета не имеют принципиального значения. У меня данный вопрос возник только при реализации метода конечного элемента. На тему скорости исполнения кода имеется отдельная [статья](#), поэтому я просто сошлюсь на нее.

Двойная интерполяция?

Многие необходимые для расчета коэффициенты приводятся в нормативных документах в форме двухмерных таблиц, в связи с чем для расчета нужно иметь соответствующую функцию. Допустим, сама таблица вбита вами в ячейки рабочей книги (как вариант, скопирована из электронной копии нормативного документа). Можно использовать некоторую функцию, которая «забирает» всю таблицу как массив (плюс два значения, при которых разыскивается искомая величина) и возвращает скаляр. См. [видео](#) с этой функцией.

```
Function doub_int(InArray, arg_vertic, arg_goris)  
' Двойная интерполяция таблицы, занесенной в Рабочую Книгу  
' (первые строка и столбец выделенного массива данных содержат аргументы  
' табулированной функции в возрастающем сверху вниз и слева направо порядке;  
' верхний левый элемент - ноль  
  
' Размеры захваченного массива  
num_of_rows = UBound(InArray, 1)  
num_of_cols = UBound(InArray, 2)  
  
' Если не выполняются необходимые условия - значение не определено  
doub_int = "n/a"  
  
' Условия для аргументов и алгоритм интерполяции  
If arg_vertic >= InArray(2,1) And arg_goris >= InArray(1,2) And arg_vertic <= InArray(num_of_rows,1) And  
arg_goris <= InArray(1,num_of_cols) Then  
i=2  
j=2  
Do  
x1 = InArray(i, 1)  
x2 = InArray(i + 1, 1)
```

```

i = i + 1
Loop Until arg_vertic >= x1 And arg_vertic <= x2
Do
y1 =InArray(1, j)
y2 =InArray(1, j + 1)
j = j + 1
Loop Until arg_goris >= y1 And arg_goris <= y2
i = i - 1
j = j - 1
a1 =InArray(i, j)
a2 =InArray(i, j + 1)
a3 =InArray(i + 1, j)
a4 =InArray(i + 1, j + 1)

If x2-x1>0 Then
res1 = a1 + (a3 - a1) * (arg_vertic - x1) / (x2 - x1)
res2 = a2 + (a4 - a2) * (arg_vertic - x1) / (x2 - x1)
End If

If y2-y1>0 Then
doub_int = res1 + (res2 - res1) * (arg_goris - y1) / (y2 - y1)
End If
End If

End Function

```

Перенос строк в текстовом отчете Выше был рассмотрен пример с выдачей значения функции в виде подробного текстового отчета с результатами. Отчет можно сделать многострочным (каждое новое значение с пояснением переносится на новую строку), используя константу vbCrLf в теле строковой переменной. В этом случае модуль с пользовательской функцией следует пометить заголовком Option Compatible.

Также в отчет вполне могут быть вставлены специальные символы — буквы греческого алфавита и т.д. (например вместо "sigma" – σ). Символы можно скопировать извне (в MS Windows – "Таблица символов", стандартные программы). **Ссылки и примеры**

Некоторые пользовательские функции для расчета металлоконструкций и калькуляции нагрузок по строительным нормам вы можете найти тут. Ниже приведен перечень ссылок на шаблоны, которые я выкладываю в общий доступ под лицензией GNU GPL v3.

1. [Коэффициент влияния формы сечения по таблице 73 СНиП II-23-81*](#)
2. [Расчет кровельных прогонов из прокатного швеллера](#)
3. [Коэффициенты для расчета ветровой нагрузки по СНиП «Нагрузки и воздействия»](#)
4. [Коэффициент для проверки общей устойчивости балок по СНиП II-23-81*](#)
5. [Коэффициент для расчета сжато-изгибаемых элементов по таблицам 74-75 СНиП II-23-81*](#)
6. [Коэффициенты для расчета центрально-сжатых стержней Расчет сварных швов крепления уголка к фасонке](#)

Вы можете скопировать код функций из перечисленных шаблонов и использовать его для обучения и практической деятельности.