

## Программный модуль для решения задач оптимального проектирования в среде SolidWorks на базе API

# 09, сентябрь 2011

авторы: Чугунов М. В., Небайкина Ю. А.

УДК 004.4

Мордовский государственный университет им. Н. П. Огарева

[m.v.chugunov@mail.ru](mailto:m.v.chugunov@mail.ru)

[juliamichch@mail.ru](mailto:juliamichch@mail.ru)

Одной из актуальных задач в области разработки программного и математического обеспечения САПР [1] является разработка приложений (добавлений) к уже существующим CAD/CAM/CAE-системам. Данный факт обусловлен следующими основными причинами:

- САПР обладают открытой архитектурой и позволяют использовать свой функционал на основе API (*Application Program Interface*), так называют набор готовых классов, функций, структур и констант, предоставляемых приложением для использования его во внешних программных продуктах [2];
- современный уровень развития САПР столь высок, что по существу исключает необходимость создания программных средств данного типа «с нуля»;
- существующие САПР нуждаются в постоянном расширении и повышении эффективности своего функционала.

Характерной особенностью современных САПР является также то, что они позволяют формировать в своих средах параметрические модели [1] проектируемых объектов. В этом случае модель однозначно определяется набором переменных, подлежащих варьированию в процессе проектирования, а некоторые изначально заданные свойства модели при этом остаются неизменными. Данное обстоятельство предполагает реализацию процесса проектирования, связанного с определением такого набора значений переменных (называемых управляемыми параметрами), которые отвечают оптимальному (наилучшему) проектному решению. Все современные САПР в той или иной форме включают в себя функционал, связанный с решением задач оптимального проектирования. Как правило, задача оптимизации ставится и решается как задача нелинейного математического программирования (НМП) в следующем виде:

найти  $\min_{\mathbf{x} \in \mathbf{R}^n} C(\mathbf{x})$  при ограничениях

$$\varphi_j(\mathbf{x}) = 0, j = \overline{1, m},$$

$$\varphi_k(\mathbf{x}) \leq 0, k = \overline{m+1, l},$$

где  $\mathbf{x}$  – вектор управляемых параметров,  $C(\mathbf{x})$  – целевая функция, выражающая собой критерий качества объекта проектирования,  $n$  – количество управляемых параметров (размерность пространства оптимизации),  $m$  – количество ограничений равенств,  $l$  – общее количество ограничений. Ограничения выражают собой геометрические свойства, физические и информационные процессы, протекающие внутри проектируемого объекта.

SolidWorks обладает мощным функционалом в части параметризации модели и анализа её свойств, а средства API дают возможность доступа к этому функционалу извне [3]. В частности, SolidWorks позволяет определять любые геометрические и массово-инерционные характеристики объекта, а также формировать аналитические функции (в терминологии SolidWorks называемые уравнениями), аргументами которых могут являться как заданные, так и вычисляемые характеристики модели. В этой связи в качестве управляемых параметров удобно использовать размеры модели, а в качестве целевой функции и функций ограничений – вычисляемые SolidWorks параметры, определяемые как непосредственно, так и на основе сформированных уравнений.

Задачи оптимизации являются весьма трудоёмкими в вычислительном отношении. Одним из подходов, позволяющих снизить вычислительные затраты, является метод многоточечных аппроксимаций [4], который состоит в поэтапной замене функций оптимизации упрощёнными аналитическими моделями, свободные параметры которых определяются по результатам экспериментов, осуществляемых с исходной «точной» моделью.

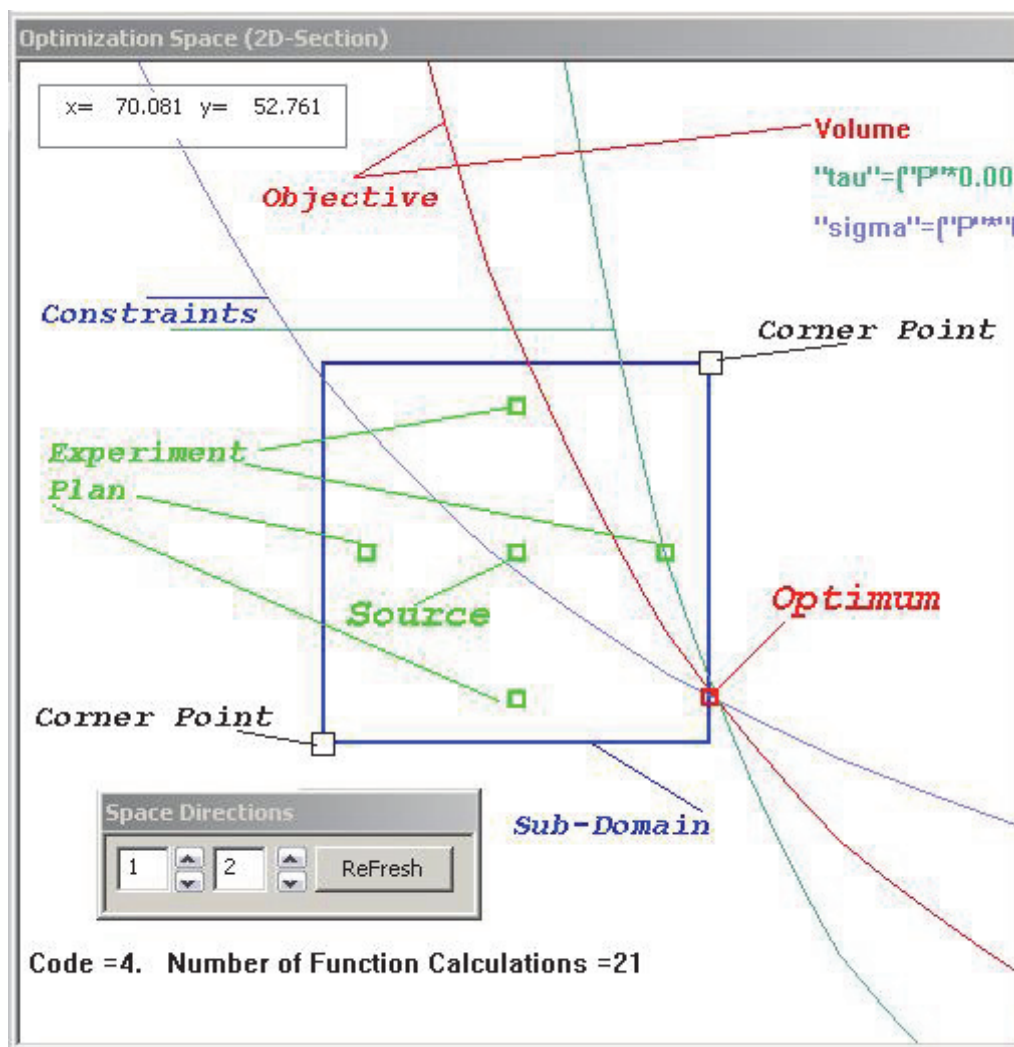


Рис. 1. Двумерное сечение пространства проектирования

Рис. 1 иллюстрирует собой процедуру построения аппроксимаций. На двумерном сечении пространства оптимизации, задаваемом номерами координатных направлений (панель Space Directions), показана подобласть поиска (синий прямоугольник Sub-Domain). План эксперимента Experiment Plan (зеленые квадратики) показан в виде креста, опорная точка креста (Source) является начальной точкой, т.е. соответствует исходному проектному решению. Кривые – функции оптимизации на данном двумерном сечении пространства проектирования: Objective – целевая функция, Constraints – функции ограничений. Вначале определяются значения функций оптимизации в точках креста средствами SolidWorks, затем, используя эти значения, взвешенным методом наименьших квадратов определяются значения свободных параметров аналитических моделей, использующихся для аппроксимаций [4]. В качестве таковых рассматриваются внутренне линейные функции: мультипликативная, обратная, корень квадратный из линейной [5]. Затем решается задача НМП для

рассматриваемой подобласти, причем, в качестве функций оптимизации используются их аппроксимации, анализ которых не связан с существенными вычислительными затратами. Красный квадратик на границе подобласти (рис. 1) – это точка, соответствующая оптимуму на этапе (Optimum). Сравнивая модельные и «точные», т.е. вычисленные средствами SolidWorks, значения функций оптимизации в этой точке, можно судить о степени адекватности упрощенной модели исходной. В случае неудовлетворительной точности, её можно повысить уменьшением размеров подобласти, или увеличением числа точек в плане эксперимента по каждому координатному направлению.

Смещение угловых точек (Corner Points) подобласти позволяет по завершению этапа перемещать подобласть в пространстве оптимизации в направлении движения от исходной точки к оптимуму. Критерием останова является принадлежность оптимальной точки внутренней части подобласти поиска совместно с относительно малыми размерами подобласти. В качестве метода решения задачи НМП на каждом этапе используется модифицированный метод последовательного квадратичного программирования [5].

Программно система оптимизации реализована в виде двух модулей, по своей архитектуре представляющие собой DLL-модули Windows. Первый из них является AddIn-модулем SolidWorks [3], функциями которого является организация COM-интерфейса [2] для доступа к функциям API SolidWorks, а также внедрение в меню Solidworks соответствующего пункта для загрузки второго DLL-модуля (Optimum.dll) (рис. 2), непосредственно реализующего систему оптимизации. Модуль Optimum.dll построен на основе двух независимых друг от друга наборов классов. Первый, аналогичный MFC набор классов и макросов, реализует пользовательский интерфейс, встроенный в окна SolidWorks и в стиле SolidWorks, второй – непосредственно реализует алгоритмы оптимизации и процедуры построения много-точечных аппроксимаций, т.е. является по существу решателем (процессором) системы. Для обмена данными между AddIn-модулем и Optimum.dll, включая указатель типа `CComPtr` на интерфейс с SolidWorks, создана специальная структура (рис. 2). Среда разработки MS Visual Studio C++, MFC, ATL.



Рис. 2. Программная реализация системы

Пример пользовательского интерфейса показан на рис. 3. В частности, на панели TaskPane размещены основные элементы управления: «дерево» задачи, включающее в себя целевую функцию (Objective), управляемые параметры (Control Variables), ограничения (Constraints). В этом же окне представлено двумерное сечение пространства оптимизации. При обращении к соответствующим элементам «дерева» в окне PropertyManager Page выводится диалоговая панель для задания необходимых параметров.

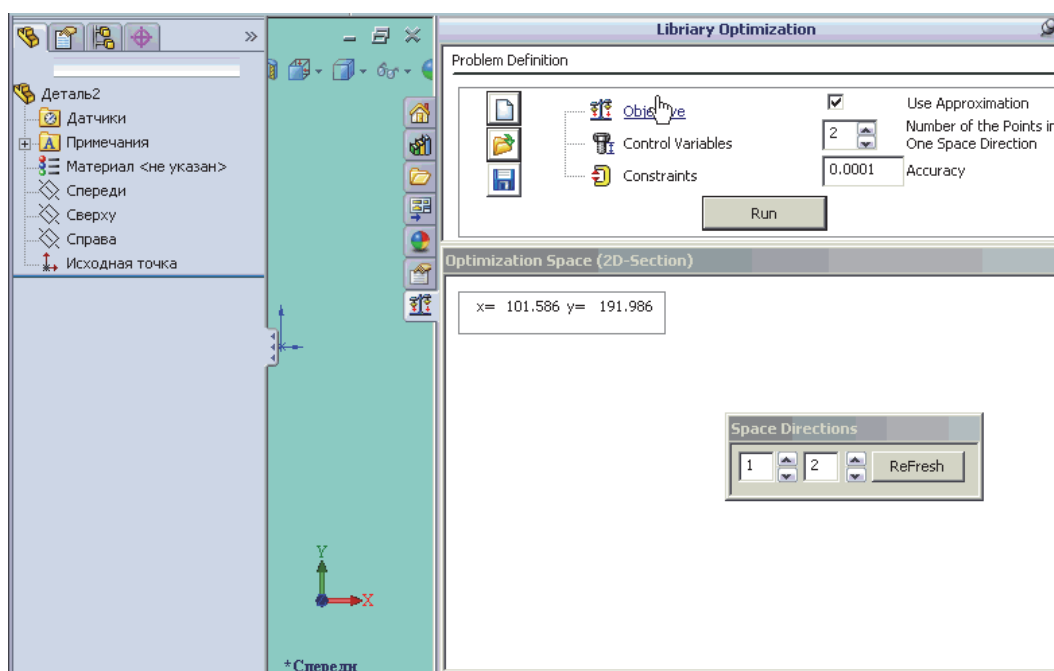


Рис. 3. Окна приложения

Рассмотрим применение данной методики на примере решения следующей задачи: максимизировать объем эллипсоида, варьируя длины всех трех его полуосей и сохраняя при этом неизменной площадь его поверхности. Рис. 4 иллюстрирует собой процедуру решения, в процессе которого пользователь перемещает подобласть поиска в различных двумерных сечениях пространства проектирования по окончании каждого этапа. При этом предусмотрена возможность зуммирования изображения, а уменьшение размеров подобласти приводит к повышению точности аппроксимаций.

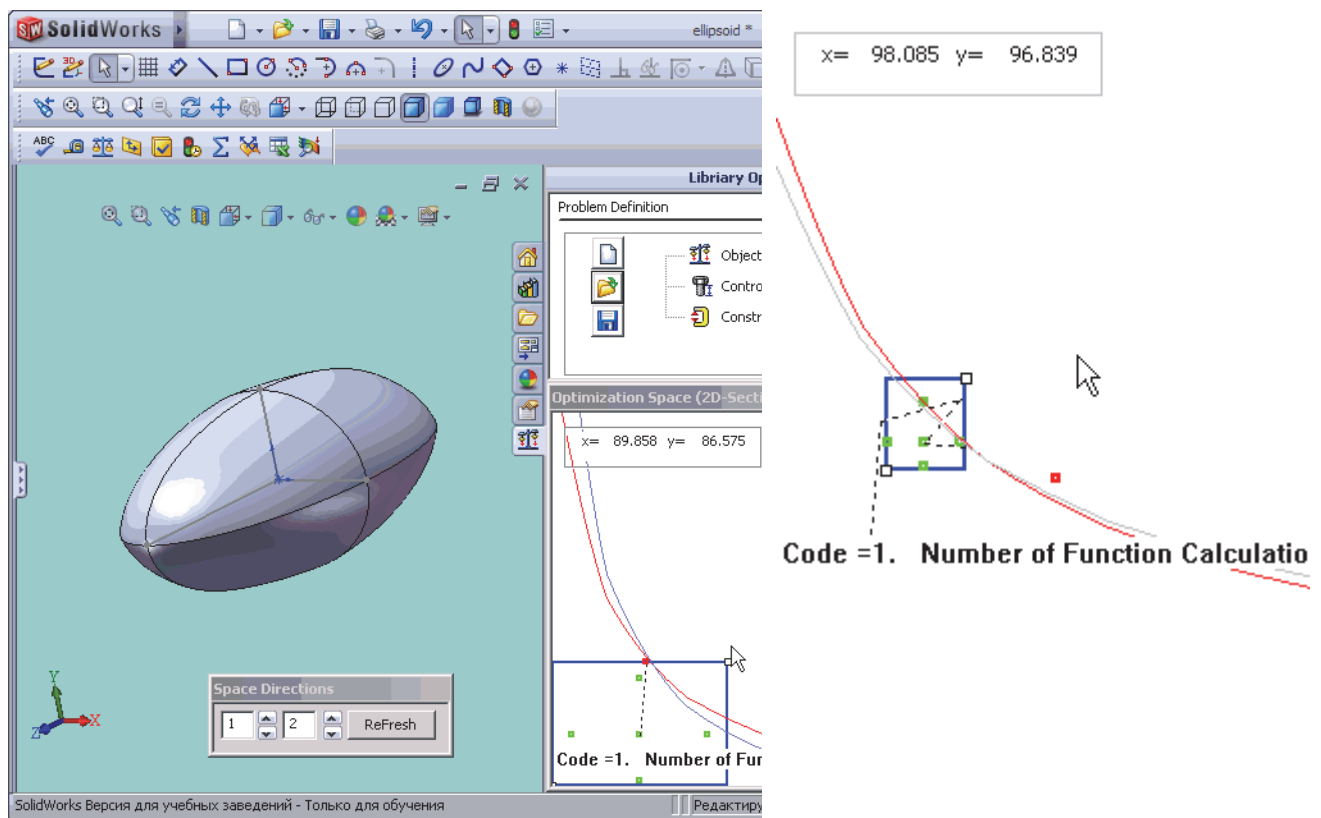


Рис. 4. Пример решения оптимизационной задачи

Как видно из рис. 5, решение сходится к шару. Если же среди функций ограничений имеются такие, которые плохо формализуются, но воспринимаются визуально (например, эстетического характера), то тогда, используя GUI [2], полезно иметь возможность менять положение оптимальной точки в пространстве оптимизации, не нарушая ранее заданных ограничений. Эта функция модуля также проиллюстрирована на рис. 5.

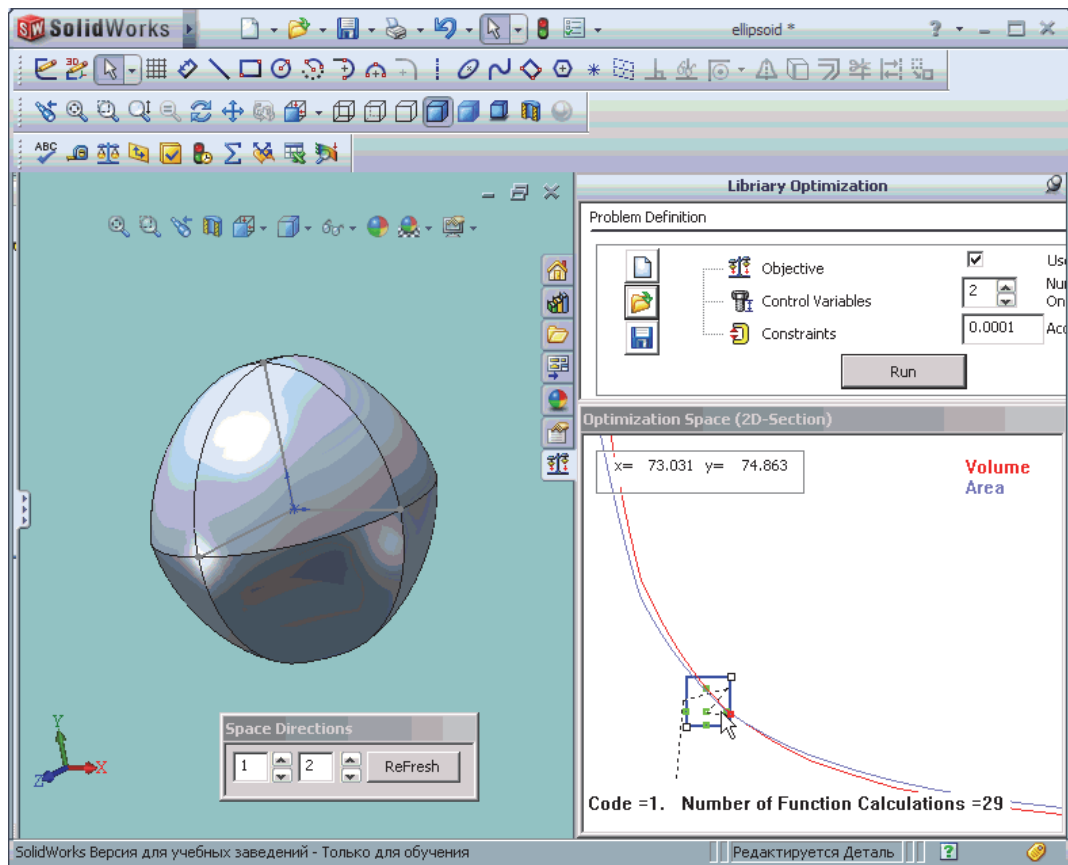


Рис. 5. Перемещение оптимальной точки в пространстве проектирования

Для тестирования разработанного программного обеспечения был решен ряд задач, традиционно рассматриваемых в курсах оптимизации в качестве тестовых (задача Штерна, Шварца, Зендора и др.) [7]. Результаты, полученные на основе рассматриваемой методики, полностью совпали с известными численными и аналитическими решениями.

На рис. 6 приведено решение известной прикладной задачи: минимизировать площадь поперечного сечения фигуры (поперечного сечения балки), ограниченной по контуру сплайном и изначально представляющей собой некоторый прямоугольник. При этом изгибающий момент и поперечная сила в сечении предполагаются неизменными, неизменными также предполагаются максимальные нормальные и касательные напряжения, действующие в сечении. В качестве управляемых параметров рассматриваются координаты узлов сплайна. На рассмотренном ранее рис. 1 представлено двумерное сечение пространства оптимизации, соответствующее одному из этапов решения этой задачи. Функции ограничений для нормальных ( $\sigma$ ) и касательных ( $\tau$ ) напряжений сформированы на основе уравнений SolidWorks. Рис. 6 иллюстрирует собой оптимальные профили при количестве узлов сплайна 3, 4 и 5 соответственно. Очевидно, что дальнейшее увеличение количества узлов сплайна все более и более будет приближать результирующий профиль к двутавру.



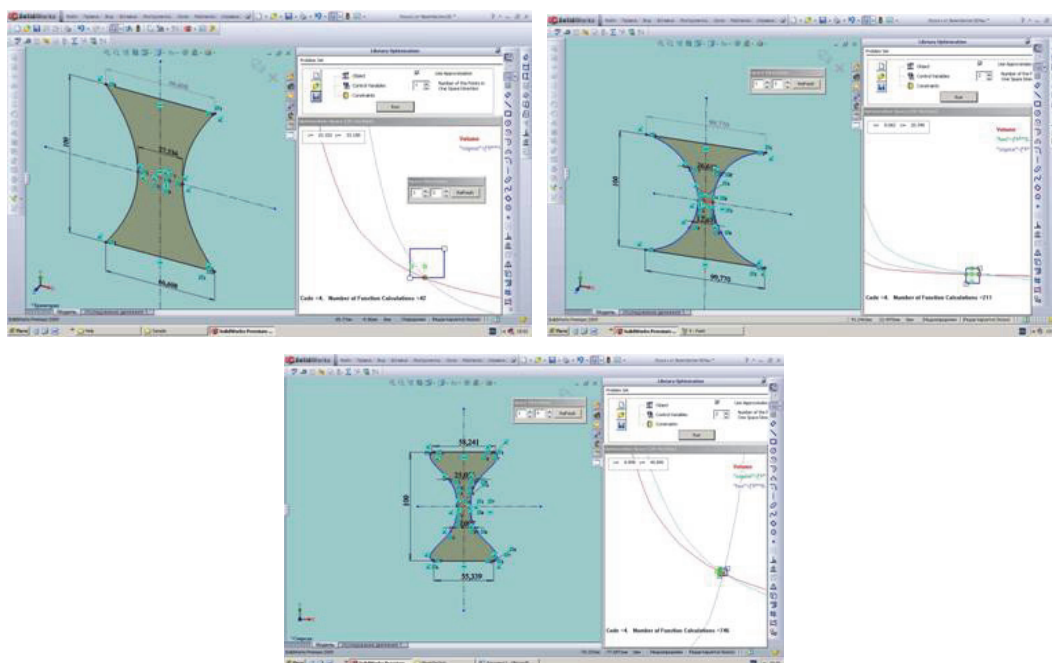


Рис. 6. Тестовая задача

Штатный функционал SolidWorks позволяет решать оптимизационные задачи, но без поэтапной аппроксимации функций оптимизации. Использование нашего модуля целесообразно в тех случаях, когда необходимо быстро получить приближенное решение, или же вычисление функций оптимизации сопряжено с большими вычислительными затратами, так, например, обстоит дело в задачах междисциплинарного типа [8].

Работа выполнена в соответствии с партнерской программой SolidWorks Corp. Partner (Research).

Литература:

1. Норенков И. П. Основы автоматизированного проектирования: учеб. для вузов. — 4-е изд., перераб. и доп.— М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. — 430 с.
2. Шеферд Дж. Программирование на Microsoft Visual C++ .NET. /Пер. с англ. - М.: Издательско-торговый дом «Русская Редакция», 2003. - 928 с.
3. Аведьян А.Б., Викентьев Е.Е. SolidWorks API – универсальная платформа для интеграции инженерных и бизнес-приложений. САПР и графика, № 6. 2006. С.32-40.
4. Toropov V.V. Multipoint approximation method in optimization problems with expensive function values. In: Sydow, A. (ed.) Computational Systems Analysis 1992, pp. 207-212, Elsevier, 1992.

5. Чугунов М.В., Небайкина Ю.А. Математическое обеспечение модуля Optimization-Works. Машиностроение: наука, техника, образование: сб.научн. тр. VIII Всеросс. научн.-практ. Конф., г.Саранск, 2010 г. – Саранск: Изд-во Мордов. Ун-та, 2010. с. 145-148.
6. Небайкина Ю.А., Чугунов М.В., Щёкин А.В. Моделирование и оптимизация объектов машиностроения в среде SolidWorks на базе API. Технологии Microsoft в теории и практике программирования. Материалы конференции/ Под ред. проф. Р.Г.Стронгина. – Нижний Новгород: Изд-во ННГУ. 2010. с. 300-304.
7. Сборник задач по оптимизации. Учебное пособие. Алексеев В.М., Галлеев Э.М., Тихомиров В.М. –М.: Наука, 1984. – 288 с.
8. Chugunov M.V., Malkov V.P., Batrakov E.N., Toropov V.V. Optimization of axisymmetric shells for flutter. In: V.V. Toropov (ed.), Proc. of 1st ASMO UK / ISSMO Conf. on Engineering Design Optimization, Ilkley, July 8-9, 1999, pp. 119-125, MCB University Press, 1999.