

## Construction of Unique Buildings and Structures



journal homepage: [www.unistroy.spb.ru](http://www.unistroy.spb.ru)



### Концепция систем препроцессора и постпроцессора для конечно-элементных расчетов пластинчато-стержневых конструкций

А.В. Алексейцев<sup>1</sup>, Н.С. Курченко<sup>2</sup>, И.В. Горбачев<sup>3</sup>, Р.М. Кашликов<sup>4</sup>

ФГБОУ ВПО Брянская государственная инженерно-технологическая академия, 241037, Россия, Брянск, проспект Станке Димитрова, 3.

#### ИНФОРМАЦИЯ О СТАТЬЕ

Финансовая поддержка:

РФФИ 13-08-00457 «Оптимизация конструктивных систем с учетом возможности аварийных ситуаций», грант губернатора Брянской области молодым ученым региона св. №01 от 05.08.2013 г.

#### История

Подана в редакцию 27 сентября 2013

#### Ключевые слова

препроцессор  
постпроцессор  
визуализация OpenGL  
сервер автоматизации COM  
библиотека визуальных компонентов  
выбор объектов  
примитивы  
дисплейные списки  
диалоговая система  
документирование

#### АННОТАЦИЯ

В статье рассмотрена концепция построения систем пре- и постпроцессорного анализа для программы конечно-элементного анализа. Системы визуализации исходных данных и результатов расчетов разрабатываются с использованием библиотеки OpenGL. Диалоговые системы строятся при помощи библиотеки визуальных компонентов (VCL). Рассматриваются отдельные аспекты реализации интерактивных свойств модели и системы документирования программного комплекса на основе технологии COM+.

#### Содержание

1. Введение	37
2. Система препроцессора	37
3. Система постпроцессора	40
4. Выбор объектов	42
5. Вывод	43

1

Контактный автор:

+7 (960) 564 3358, [aalexw@mail.ru](mailto:aalexw@mail.ru) (Алексейцев Анатолий Викторович, к.т.н, доцент, докторант)

2

+7 (920) 602 3240, [ms.kurchenko@mail.ru](mailto:ms.kurchenko@mail.ru) (Курченко Наталья Сергеевна, ассистент)

3

+7 (961) 004 9073, [hrun@yandex.ru](mailto:hrun@yandex.ru) (Горбачев Иван Владимирович, студент)

4

+7 (960) 557 4388, [rom15061506@mail.ru](mailto:rom15061506@mail.ru) (Кашликов Роман Михайлович, студент)

## 1. Введение

В настоящее время существует ряд конечно-элементных [1] программных комплексов, имеющих мощные средства пре- и постпроцессорного анализа для решения различных инженерных задач. Современные сервисные программы конечно-элементных расчетов представлены преимущественно зарубежными разработками (Femap, Patran-pre, Ansys preprocessor и т.п.), исходные коды которых недоступны для модификации. В то же время эти сервисные программы не являются совместимыми с решателями, разрабатываемыми для отдельных видов прикладных задач. Например, для расчетов, связанных со специфическими особенностями разрушения конструкций, процедурами оптимального проектирования на дискретных множествах параметров и структур деформируемых объектов.

Для автоматизации в инженерном деле получают развитие различные подходы, связанные с созданием единого информационного пространства [2, 3], автоматизации поточного выпуска документации [3], совершенствованием визуализации данных [4, 5], интеграции информации между CAD/CAE – системами [6]. Вместе с тем развиваются также и программы для решения научных и прикладных задач как многочисленными отечественными [7-10] и др., так и зарубежными разработчиками [11-13, 22, 23]. Большинство современных подходов к проектированию конечно-элементных программных комплексов требует достаточно высокой квалификации и знаний в области информационных технологий [14-15], однако, до сих пор применимы и классические способы построения сервисных программ [16-18], которые удобно использовать для решения частных научных задач. Поэтому представляется актуальным изложить концепцию разработки системы пре- и постпроцессорного анализа, которая позволит наиболее просто вводить исходные данные в решатель и удобно представлять результаты расчетов.

## 2. Система препроцессора

Базируясь на классических представлениях о сервисных программах [16-18], формируем следующие функциональные подсистемы: визуализации данных, организации диалогового режима, помощи и документирования.

**Подсистема визуализации данных** разрабатывается на основе использования аппаратно независимого низкоуровневого интерфейса OpenGL [19, 20], обращение к которому выполняется на языке Delphi [21] из программы Codegear RAD Studio 2010. Библиотека OpenGL представляет собой конечный автомат [14], работа которого описывается переменными, определяющими возможные его состояния. Такими переменными являются: текущий цвет примитивов, глубина прорисовки, содержание видовой и проекционной матриц и др. Рассмотрим примитивы (простейшие геометрические объекты), применяемые нами в системе визуализации при отображении стержневых элементов.

**Точки** используются для изображения узлов конечных элементов. Каждая точка представляется вектором вещественных чисел  $A(x, y, z, w)$ , который называется вершиной. Каждая вершина описывает положения точки в трехмерном пространстве. Координата  $w$  используется для идентификации точек, находящихся в бесконечности. При  $w > 0$  координаты точки соответствуют вектору  $A(x/w, y/w, z/w)$ . Координата  $w$  по умолчанию принимается равной единице. В случае, когда  $w = 0$ , точка находится в бесконечности и на экране компьютера не изображается.

**Отрезки.** Эти примитивы используются для отображения стержневых конечных элементов. Отрезок описывается двумя вершинами. Примером генерации отрезка может служить псевдокод, показанный на рисунке 1,а, реализация которого в OpenGL имеет вид, показанный на рисунке 1,б:

Инициализация рисования линий. Ввод координат первой точки отрезка. Ввод координат второй точки отрезка. Окончание рисования отрезка.	glBegin(GL_LINES) glVertex3f(0,0,0); glVertex3f(1,1,1); glEnd().
а	б

Рисунок 1. Реализация отображения одного отрезка

**Пластины.** Для описания пластин используются упорядоченные тройки вершин, соединяемых отрезками. Направление обхода вершин при назначении их номеров в местной системе нумерации принято по ходу часовой стрелки. Нормаль к пластине определяется векторным произведением векторов, построенных в направлениях от вершины 1 к вершине 2 и от вершины 1 к вершине 3.

Каждый из примитивов имеет следующие переменные состояния: размер точки (в пикселях), цвет, толщина изображения отрезков, сглаживание и т.п. Open GL предоставляет возможность пользователю изменять размеры узлов, толщин отрезков, обозначающих стержневые элементы, толщин кромок пластин, освещенности, цветовой палитры, масштаба нагрузок, параметров перемещения модели и др. При этом значительно улучшается восприятие отображаемых объектов, что является преимуществом по сравнению с другими системами. Например, в Femap при визуализации по умолчанию толщина стержневого элемента совпадает с толщиной пластинчатого, и цвета кромки пластины и стержня совпадают. При этом, чтобы идентифицировать стержневые элементы приходится совершать такие операции как вывод нумерации элементов, создание слоев, отображение сечений и т.п. В предлагаемой препроцессорной системе данная проблема решена.

Данный подход к визуализации позволяет работать с достаточно сложными объектами. Кроме операций поворота вокруг осей декартовой системы координат, масштабирования и панорамирования изображения конструкции, используется пять источников света. Один из них расположен на значительном удалении от модели и служит фоновым освещением. Остальные источники располагаются вокруг конечно-элементной модели по одному в каждом из квадрантов объема визуализации (рисунок 2).

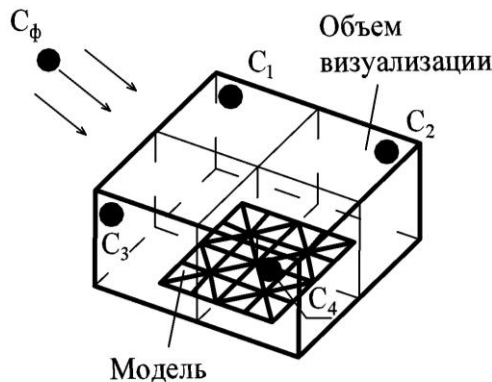


Рисунок 2. Схема освещения в пространстве визуализации:  $C_\phi$  – фоновый источник;  $C_1...C_4$  – точечные источники

Отображение одинаковых примитивов осуществляется с помощью дисплейных списков [21], применение которых позволяет быстро показывать большие группы этих объектов. Пример использования дисплейного списка приведен на рисунке 3.

<p>Генерируем дисплейный список «points».                  Открываем этот список.                  Устанавливаем размер точек.                  В цикле по числу точек выполняем:                  инициализацию блока точек;                  отрисовываем iuz точек в 3D;                  закрываем блок точек и цикл.                  Закрываем дисплейный список.</p>	<pre>points := glGenLists (1); glNewList (points, GL_COMPILE); glPointSize(0.8); for z:=1 to iuz do begin glbegin(gl_Points); glvertex3f(x, y, z); glend; end; glEndList;</pre>
а	б

Рисунок 3. Реализация дисплейного списка для отображения группы отрезков

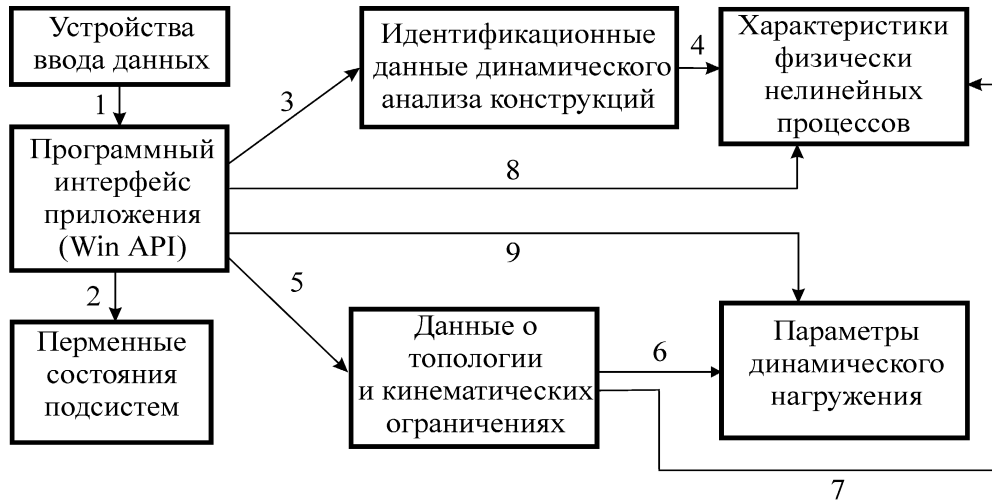
**Диалоговая система ввода исходных данных.** Построение этой системы выполняется с помощью библиотеки визуальных компонентов (VCL) [5]. В диалоговой системе реализован режим обмена информацией между человеком и вычислительной машиной при помощи диалоговых окон [1, 4]. Фрагмент схемы ввода информации представлен на рисунке 4. Каждый исполняемый блок этой схемы состоит из одного или нескольких окон. Изменение переменных состояния, происходящее при использовании этих окон, записываются в реестр операционной системы. Пример диалогового окна, реализующего блок «Параметры динамического нагружения» изображен на рисунке 5. С помощью данного окна выполняются группы операций 1 и 8, 1 и 9 (см. рисунок. 4). Каждое диалоговое окно снабжено справочной системой и имеет средства защиты от ввода неправильных данных.

Диалоговая система содержит следующие элементы:

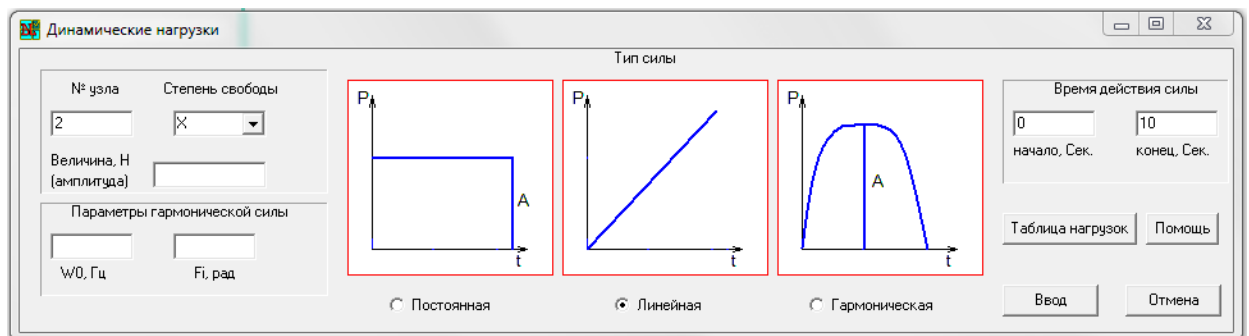
- средства ввода данных о конечно-элементной модели и типе расчета;

- ввод диаграмм деформирования для каждого элемента;
- задание параметров динамического процесса (время интегрирования, тип динамической нагрузки, время действия сил и т.п.).

**Система помощи и документирования.** Система помощи (справка) представлялась набором html-файлов, созданных программой HtmlWorkshop. В результате компиляции html-файлов создавался файл справки FEM.chm, который ассоциировался с рабочими формами.



**Рисунок 4 . Функциональные блоки диалоговой системы препроцессора: 1-9 – операции передачи данных о конечно-элементной модели**



**Рисунок 5. Окно ввода динамических нагрузок**

Документирование введенных исходных данных выполняется на основе технологии component object model (COM+). При формировании системы документирования использовалась клиент-серверная модель. Серверами автоматизации называются программы, которыми можно управлять через интерфейс COM+. Клиентом считается программа, вызывающая сервер автоматизации. В предлагаемом препроцессоре документирование может быть реализовано пользователем при сохранении данных о конечно-элементной модели. Сервером автоматизации документирования мы выбрали приложение MS WORD. Схема включения сервера показана на рисунке 6. Фрагмент отчета, сформированного с применением описанного подхода, изображен на рисунке 7.

<p>Инициализируем приложение. Связываем сервер автоматизации с активным файлом WORD. Устанавливаем форматирование текста по умолчанию. Центрируем текст.</p> <p>Пишем в файле 'BGITAFEM v1.0'. Выделяем написанный шрифт. Устанавливаем параметры шрифта. Записываем файл WORD. Закрываем файл. Запускаем записанный файл для просмотра пользователем. Отключаем сервер автоматизации.</p>	<pre>WordApplication1.Documents.Add( ); form1.Worddocument1.ConnectTo (ActiveDocument); WordParagraphFormat1.ConnectTo (Selection.ParagraphFormat); WordParagraphFormat1.Alignment:= wdAlignParagraphCenter; Selection.TypeText('BGITAFEM v1.0'); WordDocument1.Range(a,b).select; WordDocument1.Range(a,b).Font.Bold:=1; form1.Worddocument1.SaveAs(n); form1.Worddocument1.Close; WordApplication1.Visible:=true; WordApplication1.Documents.Open (n,...); WordApplication1.Disconnect;</pre>
--	---

Рисунок 6. Пример использования сервера автоматизации MS WORD

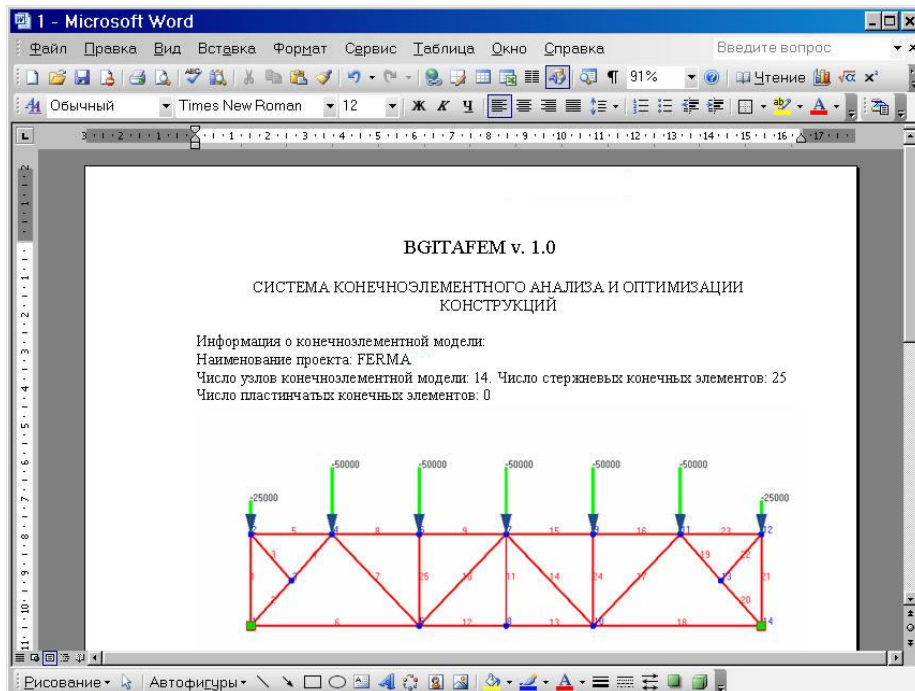


Рисунок 7. Пример использования сервера автоматизации MS WORD

Кроме документирования данных предусмотрена совместимость препроцессора с другими приложениями путем сохранения данных в формате dxf.

### 3. Система постпроцессора

Постпроцессор включает в себя систему визуализации результатов расчетов и диалоговую систему представления выходных данных (рисунок 8). Имеются следующие подсистемы:

- построения эпюр внутренних усилий в стержнях и напряжений в сечениях для стержневых конечных элементов;
- построения полей напряжений для пластинчатых конечных элементов;
- построения цветовой индикации перемещений и деформированных схем объектов;
- вывода результатов расчетов на экран.



Рисунок 8. Компоненты системы представления результатов расчетов

Поясним порядок работы постпроцессора (см. рисунок 8). Значения усилий и перемещений, вычисленные решателем направляются в блок 1. Далее из блока 1 выполняется их считывание и отсылка в блоки 4–6. В этих блоках формируются массивы объектов-записей, размерность которых определяется числом конечных элементов конструкции. Каждый элемент такого массива в блоке 4 представляет собой для стержневого конечного элемента структуру, показанную на рисунке 9.

Cmb: integer;	Номер комбинации нагружений.
Uz1: integer;	Номер начального узла.
Uz2: integer;	Номер конечного узла.
N1: real;	
N2: real;	
Qz1: real;	
Qz2: real;	
Qy1: real;	Усилия N, Qz, Qy, Mx, My, Mz в
Qy2: real;	узловых сечениях стержня
Mz1: real;	
Mz2: real;	
My1: real;	
My2: real;	
Mx1: real;	
Mx2: real;	

Рисунок 9. Данные о результатах расчета стержня

- инициализация групп вершин и цветов, содержащих данные об базовых вершинах и формируемых промежуточных вершинах. Вычисляются промежуточные значения RGB цветов;
- визуализация объектов с использованием данных из сформированных на предыдущем этапе массивов.

Пример визуализации эпюры продольных сил представлен на рисунке 10.

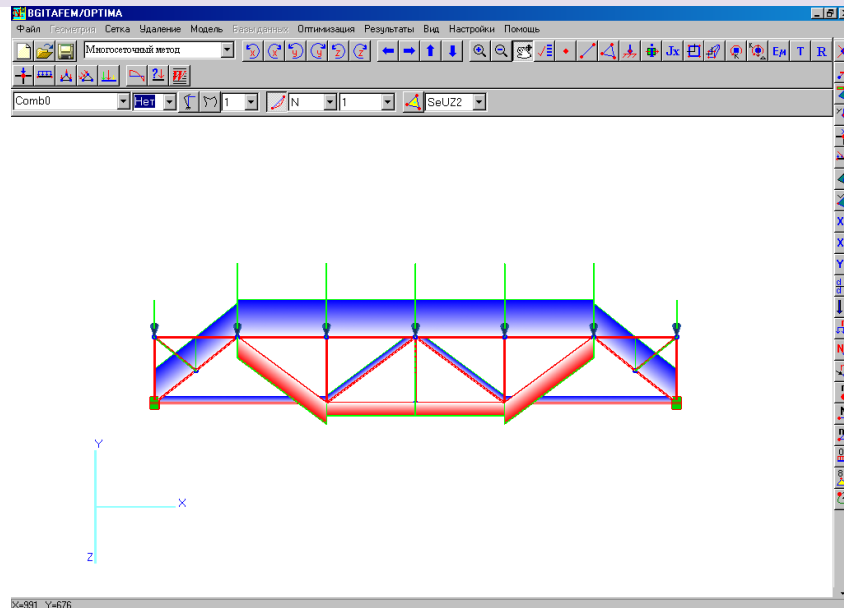


Рисунок 10. Общий вид эпюры продольных сил в ферме

Функция блока 6 (см. рисунок 8) – визуализации деформированного состояния объекта. Вычисленные в результате конечно-элементного анализа перемещения узлов являются соответствующими приращениями координат для изображения деформированной схемы. Пример деформированной схемы коробчатой балки представлен на рисунке 10.

#### 4. Выбор объектов

Для расширения возможностей систем как препроцессорного, так и постпроцессорного анализа предусматривается возможность выбора объектов пользователем. Первоначально все рисуемые на экране объекты отображаются в буфер кадра. Далее включается режим выбора, и кадр рисуется заново. При этом каждому из примитивов, пересекающих видимый объем, определяемый текущим состоянием видовой и проекционной матриц [22], присваивается уникальное имя. Все имена загружаются в специальный стек, при обращении к которому можно определить какие примитивы находятся в области просмотра (на экране). Механизма выбора реализуется путем выполнения ряда этапов:

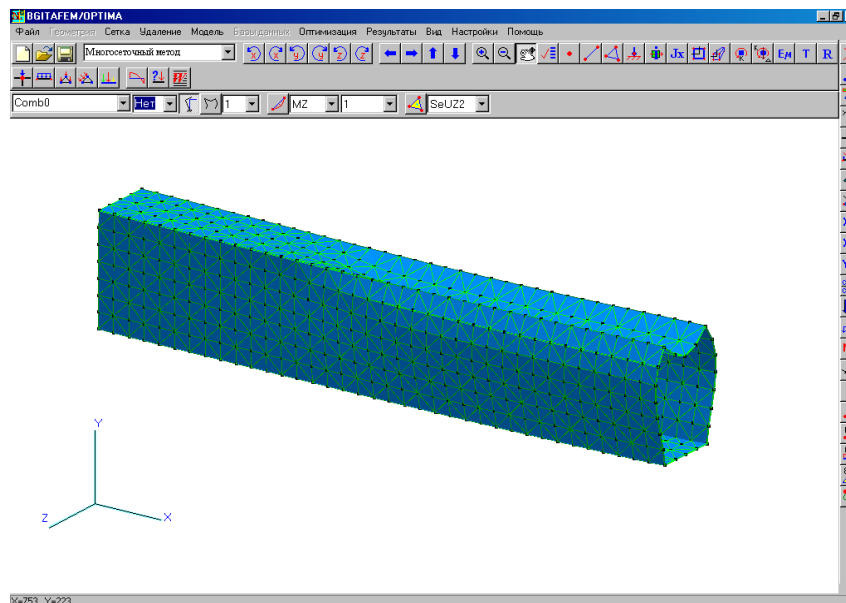


Рисунок 10. Деформированная схема пластинчатой системы

1. Определить массив, который будет использоваться идентификации видимых примитивов.

2. Включить режим выбора и инициализировать стек имен.
3. Определить объем  $V$ , который предполагается использовать для выбора.
4. Загрузить в стек имен уникальное имя примитива и воспроизвести примитив.
5. Выключить режим выбора и проанализировать имена примитивов, попавших в объем  $V$ .

По аналогичной схеме реализуется и отбор элементов. Отбор выполняется при определении малого объема  $V$ , расположенного в области нахождения курсора мыши. При этом необходимо при каждой операции отбора сохранять текущую матрицу проецирования в матричном стеке.

Для представляемого комплекса разработана схема, при которой выбор объектов реализован по принципу, используемому в AutoCAD: размер объема  $V$  определяется площадью прямоугольника, получаемого при захвате области просмотра мышью и глубиной прорисовки сцены визуализации. Если в процессе работы над моделью пользователь формирует группы элементов с одинаковыми свойствами (стержни одного поперечного сечения), то можно выделять эти группы, используя операции выбора или отбора.

Можно отметить, что при создании гибких схем выбора объектов работа с конечно-элементной моделью значительно упрощается. В частности для выбранной группы элементов можно в 1-2 операции изменить их видимость, цвет, толщину, номер типа сечения, в то время как в других системах, например в femap, для этого может понадобиться до 10 операций.

## 5. Вывод

Предложена концепция систем пре- и постпроцессорного анализа для конечно-элементного моделирования стержневых и пластинчато-стержневых систем. Используемые процедуры позволяют выполнять относительно простое и удобное интерактивное сопровождение конечно-элементных расчетов. Разработанные алгоритмы будут способствовать решению исследовательских задач, связанных оценкой напряженно-деформированного состояния конструкций. Представляемые процедуры реализованы для программного комплекса BGITAFEM/OPTIMA [8] и могут быть использованы для других процессоров в рамках решения различных научных задач.



## Литература

1. Zienkiewicz O.C., Taylor R. L., Zhu J.Z. The Finite Element Method: its Basis and Fundamentals. Seventh Edition. 2013. 756 p.
2. Воробьев А. Данилова Л. Игнатов Б. Сценарий и механизмы создания единого информационного пространства // CADmaster. 2010. № 5. С. 48-51.
3. Рындин А., Галкина О., Благодарь А. Автоматизация потоков документации – важный шаг к созданию ЕИП // REM. 2012. № 4. С. 42-48.
4. Сладковский А., Кузьмин Е., Шалаева О. Информационная система визуализации 3D-моделей на базе Intergraph SmartPlant Review // САПР и графика. 2011. № 9. С. 25.
5. Cheney D.C. A user interface specification for CAD/CAM Application Development. Dept. of Mechanical engineering Brigham Young University, 1994. 144 p.
6. Долгалев Д. Обмен данными между различными системами // САПР и графика. 2011. № 9. С. 73-75.
7. Алексейцев А.В., Серпик И.Н. Препроцессорный и постпроцессорный анализ в расчетах и оптимальном проектировании несущих систем // Вклад ученых и специалистов в национальную экономику: сб. науч. тр. к конф. Брянск: БГИТА, 2005. С. 219-222.
8. Алексейцев А.В. Использование современных информационных технологий в оптимизации несущих систем // Перспективы развития новых технологий в строительстве и подготовке инженерных кадров в Республике Беларусь: матер. XV Междунар. науч.-метод. семинара. Полоцк: ПГУ, 2008. Ч.2. С. 259-263.
9. Войтенко В.И. Программный продукт «Программный комплекс Sapr\_100\_W». Свидетельство автора №7456 от 18.04.03.
10. Серпик И.Н., Алексейцев А.В. Программа для ЭВМ BGITAFEM/OPTIMA Свидетельство о государственной регистрации программы для ЭВМ № 2012661425 выдано 14.12.2012.
11. A complete development process of finite element software for body-in-white structure with semi-rigid beams in .NET framework / Wenjie Zuo, Weiwei Li, Tao Xu, Shengyi Xuan, Jingxin Na. // Advances in Engineering Software. 2012. Vol. 45. Issue 1. Pp. 261-271.
12. Automatic FE modeler using stiffener-based mesh generation algorithm for ship structural analysis / Beom-Seon Jang, Yong-Suk Suh, Eun-Ki Kim, Tae-Hee Lee // Marine Structures. 2008. Vol. 21. Issues 2–3. Pp. 294-325.
13. Kiran H. Shivanna, Srinivas C. Tadepalli, Nicole M. Grosland Feature-based multiblock finite element mesh generation // Computer-Aided Design. 2010. Vol. 42. Issue 12. Pp. 1108-1116.
14. Кельтон В., Лоу А. Имитационное моделирование. Классика CS. СПб.: Питер, 2004. 848 с.
15. Телешевский В.И. О концепции измерительной информатики // Тр. III международного конгресса "Конструкторско-технологическая информатика" М.: Станкин, 1996. С.134-135.
16. Хокс Б. Автоматизированное проектирование и производство. Пер. с англ. М.: Мир, 1991. 296 с.
17. Грувер М., Зиммерс Э. САПР и автоматизация производства. Пер. с англ. М.: Мир, 1987. 528 с.
18. Геммерлинг Г.А. Система автоматизированного проектирования стальных конструкций. М.: Стройиздат, 1987. 210 с.
19. Ву М., Нейдер Д., Девис Т., Шрайнер Д. OpenGL. Официальное руководство программиста. СПб: ООО «ДиаСофтЮП», 2002. 592 с.
20. Хилл Ф. OpenGL. Программирование компьютерной графики. Для профессионалов. СПб: Питер, 2002. 1088 с.
21. Фаронов В. Система программирования Delphi. СПб.: БХВ-Перербург, 2004. 912 с.
22. Consistent Finite Element mesh generation for meso-scale modeling of textile composites with preformed and compacted reinforcements / Grail G., Hirsekorn M., Wendling A., Hivet G., Hambli R. //Composites Part A: Applied Science and Manufacturing. 2013. Vol. 55. Pp. 143-151.
23. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions / Mantiuk R., Kim KJ., Rempel AG., Heidrich W. // ACM Transactions on Graphics (TOG) - proceedings of ACM SIGGRAPH-2011. 2011. Vol. 30(11). Pp. 1 -14.

## Concept of the preprocessor and postprocessor for finite element analysis a plate-beam structures

A.V. Alekseytsev<sup>5</sup>, N.S. Kurchenko<sup>6</sup>, I. V. Gorbachev<sup>7</sup>, R. M. Kashlikov<sup>8</sup>

*Bryansk State Academy of Engineering and Technology, 3 Stanke Dimitrova avenue, Bryansk, 241037, Russia.*

### ARTICLE INFO

Financial support:

RFBR 13-08-00457 "Optimization of structural systems with the possibility of emergencies," Governor of the Bryansk region grant for young scientists in the region (№ 01, 05.08.2013)

### Article history

Received 27 September 2013

### Keywords

preprocessor  
postprocessor  
OpenGL rendering  
COM automation server  
visual components library  
choice of objects  
primitives  
display lists  
dialogue system  
documentation

### ABSTRACT

The article considers a concept of building a system of pre-and post-processing to program of the finite element analysis. Rendering system of input data and calculation results are developed using the OpenGL interface. Interactive systems are built using Visual Component Library (VCL). Separate aspects of the interactive properties of the model and system documentation software based on the COM+ technology is given.

---

*Corresponding author:*

- 5 +7 (960) 564 3358, aalexw@mail.ru (Anatoly Viktorovich Alekseytsev, Ph.D, associate professor)  
6 +7 (920) 602 3240, ms.kurchenko@mail.ru (Natalya Sergeevna Kurchenko, assistant)  
7 +7 (961) 004 9073, hpun@yandex.ru (Ivan Vladimirovich Gorbachev, student)  
8 +7 (960) 557 4388, rom15061506@mail.ru (Roman Mikhailovich Kashlikov, student)

## References

1. Zienkiewicz O.C., Taylor R. L., Zhu J.Z. The Finite Element Method: its Basis and Fundamentals. Seventh Edition. 2013. 756 p.
2. Vorob'yev A., Danilova L., Ignatov B. *Stsenariy i mekhanizmy sozdaniya yedinogo informatsionnogo prostranstva* [Scenario and mechanisms of common information space] // CADmaster. 2010. № 5. Pp. 48-51. (rus)
3. Ryndin A., Galkina O., Blagodyr' A. *Avtomatizatsiya potokov dokumentatsii – vazhnyy shag k sozdaniyu YEIP* [Automating the document flow - an important step towards the creation of the CIS] // REM. 2012. № 4. Pp. 42-48. (rus)
4. Sladkovskiy A., Kuz'min Ye., Shalayeva O. *Informatsionnaya sistema vizualizatsii 3D-modeley na baze Intergraph SmartPlant Review* [Information System Visualization 3D - based models Intergraph SmartPlant Review] // SAPR i grafika. 2011. № 9. Pp. 2-5. (rus)
5. Cheney D.C. A user interface specification for CAD/CAM Application Development. Dept. of Mechanical engineering Brigham Young University, 1994. 144 p.
6. Dolgalev D. *Obmen dannymi mezhdu razlichnymi sistemami* [Exchange of data between different systems] // SAPR i grafika. 2011. № 9. Pp. 73-75. (rus)
7. Alekseytsev A.V., Serpik I.N. *Preprotsessornyy i postprotsessornyy analiz v raschetakh i optimal'nom proyektirovaniy nesushchikh system* [Preprocessor and postprocessor analysis calculation and optimal design of bearing systems] // *Vklad uchenykh i spetsialistov v natsional'nyu ekonomiku: sb. nauch. tr. k konf. Bryansk: BGITA*, 2005. Pp. 219-222. (rus)
8. Alekseytsev A.V. *Ispol'zovaniye sovremennykh informatsionnykh tekhnologiy v optimizatsii ne-sushchikh sistem* [The use of modern information technology to optimize bearing systems] // *Perspektivy razvitiya novykh tekhnologiy v stroitel'stve i podgotovke inzhenernykh kadrov v Respublike Belarus': mater. XV Mezhdunar. nauch.-metod. seminara. Polotsk: PGU, 2008. Part.2*. Pp. 259-263. (rus)
9. Voytenko V.I. *Programmnyy produkt «Programmnyy kompleks Sapr\_100\_W»* [The software product "software package Sapr\_100\_W"]. *Svidetel'stvo avtora №7456 ot 18.04.03*. (rus)
10. Serpik I.N., Alekseytsev A.V. *Programma dlya EVM BGITAFEM/OPTIMA* [The computer program BGITAFEM / OPTIMA] *Svidetel'stvo o gosudarstvennoy registratsii programmy dlya EVM № 2012661425 vydano 14.12.2012*. (rus)
11. A complete development process of finite element software for body-in-white structure with semi-rigid beams in NET framework / Wenjie Zuo, Weiwei Li, Tao Xu, Shengyi Xuan, Jingxin Na. // *Advances in Engineering Software*, Vol. 45. Issue 1, March 2012. Pp. 261-271.
12. Automatic FE modeler using stiffener-based mesh generation algorithm for ship structural analysis / Beom-Seon Jang, Yong-Suk Suh, Eun-Ki Kim, Tae-Hee Lee // *Marine Structures*. 2008. Vol. 21. Issues 2–3. Pp. 294-325.
13. Kiran H. Shivanna, Srinivas C. Tadepalli, Nicole M. Grosland Feature-based multiblock finite element mesh generation // *Computer-Aided Design*. 2010. Vol. 42. Issue 12. Pp. 1108-1116.
14. Kel'ton V., Lou. A. *Imitatsionnoye modelirovaniye. Klassika CS.*[Simulation. Classic CS.] *SPb.: Piter*, 2004. 848 p. (rus)
15. Teleshevskiy V.I. *O kontseptsii izmeritel'noy informatiki* [On the concept of measurement science] // *Tr. III mezhdunarodnogo kongressa "Konstruktorsko-tekhnologicheskaya informatika."* M.: Stankin, 1996. Pp.134-135. (rus)
16. Khoks B. *Avtomatizirovannoye proyektirovaniye i proizvodstvo. Per. s angl.* [Computer-aided design and manufacturing.] M.: Mir, 1991. 296 p. (rus)
17. Gruver M., Zimmers E. *SAPR i avtomatizatsiya proizvodstva. Per. s angl.* [CAD and manufacturing automation] M.: Mir, 1987. – 528 p. (rus)
18. Gemmerling G.A. *Sistema avtomatizirovannogo proyektirovaniya stal'nykh konstruksiy.*[Computer-aided design of steel structures] M.: Stroyizdat, 1987. 210 p. (rus)
19. Vu M., Neyder D., Devis T., Shrayner D. *OpenGL. Ofitsial'noye rukovodstvo programmista.*[OpenGL. Official Programmer's manual] *SPb: OOO «DiaSoftYUP»*, 2002. 592 p. (rus)
20. Khill F. *OpenGL. Programirovaniye komp'yuternoy grafiki. Dlya professionalov.* [Programming computer graphics. For professionals.] *SPb: Piter*, 2002. 1088 p. (rus)
21. Faronov V. *Sistema programirovaniya Delphi.* *SPb.: BKHV-Petersburg*, 2004. 912 p. (rus)

22. Consistent Finite Element mesh generation for meso-scale modeling of textile composites with preformed and compacted reinforcements / Grail G., Hirsekorn M., Wendling A., Hivet G., Hambli R. // Composites Part A: Applied Science and Manufacturing. 2013. Vol. 55. Pp. 143-151.
23. Mantiuk R., Kim K.J., Rempel AG., Heidrich W. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions / Mantiuk R., Kim K.J., Rempel AG., Heidrich W. // ACM Transactions on Graphics (TOG)—proceedings of ACM SIGGRAPH-2011. 2011. Vol. 30(11). Pp. 1–14.