

АРХИТЕКТУРА И СТРОИТЕЛЬСТВО

УДК 624.012.03

П. И. Сафронов

ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА CODE_ASTER ДЛЯ РЕШЕНИЯ ЗАДАЧ СТРОИТЕЛЬНОЙ МЕХАНИКИ И ТЕОРИИ УПРУГОСТИ

Рассматриваются основные возможности программного комплекса Code_Aster для расчёта сложных механических моделей методом конечных элементов. Рассматривается возможность использования программного комплекса Code_Aster при изучении дисциплин «Строительная механика» и «Теория упругости». Предлагается последовательность этапов по внедрению программного комплекса Code_Aster в учебный процесс.

Ключевые слова: метод конечных элементов, программный комплекс Code_Aster, строительная механика, теория упругости.

Строительство является одной из ведущих отраслей современной российской экономики и требует подготовки большого количества высококвалифицированных кадров. Инженер-строитель должен владеть знаниями и навыками, позволяющими ему решать профессиональные задачи на высоком уровне в условиях жёсткой конкуренции. При подготовке таких специалистов большую роль играет блок дисциплин механики твёрдых тел. Эти дисциплины выступают в качестве базиса, на который опирается изучение последующих специальных дисциплин. Среди дисциплин механики твёрдых тел можно выделить «Строительную механику» и «Теорию упругости» в силу их фундаментальности и универсальности.

Во второй половине XX века положение в области строительной механики и теории упругости кардинальным образом изменилось. Как отмечается в [1], после появления ЭВМ эти дисциплины в значительной степени стали экспериментальными науками, так как стало возможно ставить и выполнять вычислительные эксперименты. Такое кардинальное изменение в основном связано с двумя обстоятельствами.

Первое обстоятельство заключается в бурном развитии средств вычислительной техники и персональных компьютеров в частности. В соответствии с законом Мура, вычислительная мощность компьютера удваивается в среднем за полтора года. В результате типичный персональный компьютер настоящего времени по вычислительным возможностям на порядки превосходит огромный вычислительный центр середины 80-х годов прошлого века. Во многих современных образовательных и научных учреждениях студенты, аспиранты и исследователи могут использовать кластеры и суперкомпьютеры. Интенсивно развиваются облачные технологии, которые позволяют получить доступ к огромным вычислительным ресурсам.

Второе обстоятельство заключается в появлении и быстром развитии численных методов решения задач строительной механики и теории упругости. В настоящее время очевидно, что лидирующую роль среди численных

методов занял метод конечных элементов. Суть этого метода заключается в дискретизации рассматриваемой области расчётной сеткой на элементы конечных размеров и аппроксимации искомой функции системой кусочно-непрерывных функций. За последние несколько десятилетий метод конечных элементов прошёл путь от специализированного метода расчёта конструкций до универсального метода решения задач, описываемых системами уравнений в частных производных.

Вышеперечисленные обстоятельства требуют изменения методики изучения строительной механики и теории упругости в технических вузах, осуществляющих подготовку инженеров-строителей. Методика изучения этих дисциплин обязательно должна включать использование современных программ или программных комплексов, реализующих метод конечных элементов. При этом возникает проблема выбора программного комплекса из многообразия существующих вариантов.

На сегодняшний день существует большое количество программ и программных комплексов, предназначенных для решения разнообразных задач методом конечных элементов. Все эти программы и программные комплексы можно разделить на три класса.

К первому классу относятся программы и программные комплексы, распространяемые на коммерческой основе, обладающие широким спектром возможностей и имеющие официальные сертификаты соответствия нормативным требованиям. К программам и программным комплексам этого класса относятся следующие продукты: ABAQUS, ADINA, ANSYS, DIANA, COSMOS, GTSTRUDL, NASTRAN, SCAD, Straus7, ПК ЛИРА и другие.

Представителями второго класса являются свободно распространяемые программы и программные комплексы, обладающие более узким спектром возможностей, не имеющие официальных сертификатов и разработанные одним автором или небольшой группой авторов. Часто программы этого класса являются приложением к печатному изданию. В качестве примера можно привести программы RADIUS [2] и FEPC [3]. Из программ второго класса особого внимания заслуживает программа FEAPrv. Эта программа создана Робертом Тейлором (R. L. Taylor), профессором инженерно-строительного факультета Беркли, Калифорния, США. Впервые концепция программы и исходный текст основных модулей были опубликованы в [4]. Готовая к выполнению программа впервые стала доступна пользователям как составная часть [5] и [6]. Программа FEAPrv непрерывно развивается с 1976 года. Большой вклад в развитие программы внесли: профессор Juan C. Sumo (стратегия решения нелинейных задач, учёт конечных деформаций для объёмных элементов на основе функционалов Ху-Васидзу), профессор Adnan Ibrahimbegovich (большие перемещения балок и оболочек), профессор Sanjay Govindjee (динамическое распределение массивов и поддержка параллельных вычислений на многопроцессорных системах) [7]. С 1976 года общее идейное руководство разработкой этой программы осуществляет профессор Olek Zienkiewicz.

К третьему классу относятся программы и программные комплексы, занимающие промежуточное положение между первым и вторым классами. С

одной стороны, эти программы и программные комплексы по своим возможностям приближаются к коммерческим продуктам и развиваются коммерческими организациями или группами таких организаций. С другой стороны, эти программы и программные комплексы распространяются на свободной основе и часто не имеют всех необходимых сертификатов. Достойным представителем продуктов третьего класса является французский программный комплекс Code_Aster (www.code-aster.org).

Программный комплекс Code_Aster появился в 1991 году. Разработчиком этого комплекса выступила французская фирма «EDFR&D», которая занимается исследованиями в области ядерной энергетики. До 2001 года программный комплекс Code_Aster распространялся на коммерческой основе. С 2001 года программный комплекс Code_Aster распространяется как открытый программный продукт на условиях лицензии GNUFDL (www.gnu.org/copyleft/fdl.html). Переход на платформу открытого программного обеспечения преследовал следующие цели:

- улучшение качества и надёжности программного продукта;
- увеличение известности среди пользователей;
- распространение знаний и навыков по моделированию сложных систем;
- поддержка становления промышленных и академических сообществ;
- формирование кадрового резерва из студентов и аспирантов, умеющих работать с этим программным продуктом.

В настоящее время доступны 4 версии Code_Aster:

1. Стабильная версия. Эта же версия используется фирмой «EDFR&D» для исследований в области ядерной энергетики. На данный момент это версия 10.7. Стабильная версия обновляется каждые 6 месяцев.

2. Стабильная версия с обновлениями. Эта версия представляет собой промежуточную версию между двумя стабильными версиями, в которой исправлены обнаруженные ошибки. Эта версия обновляется 1 раз в месяц.

3. Тестовая версия. Представляет собой копию нестабильной исследовательской версии. Эта версия обновляется 1 раз в 6 месяцев.

4. Нестабильная исследовательская версия. Отражает текущее состояние Code_Aster в процессе добавления новых возможностей. Эта версия обновляется 1 раз в неделю.

Программный комплекс Code_Aster работает под большинством современных 64-х разрядных операционных систем семейства Linux. Компания «EDFR&D» официально не поддерживает работу Code_Aster под операционной системой MSWindows.

В русскоязычной технической литературе сложилась схема разделения программных комплексов, реализующих метод конечных элементов, на отдельные функциональные части, которые показаны на рисунке 1.

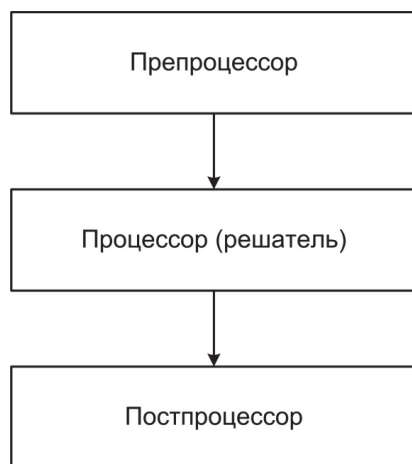


Рис. 1. Структура программного комплекса, реализующего метод конечных элементов

Препроцессор — это часть, в которой определяются геометрия и топология рассматриваемой области, характеристики среды, условия закрепления, внешние воздействия.

Процессор (решатель) — это часть, в которой формируются матрицы для отдельных конечных элементов, формируются глобальные матрицы для модели в целом, формируются системы разрешающих уравнений для модели в целом, решаются системы разрешающих уравнений, формируются массивы результатов расчёта для отдельных конечных элементов и модели в целом.

Постпроцессор — это часть, которая отвечает за отображение результатов расчётов в графической форме и формирование различных отчётов с результатами расчётов.

С точки зрения вышеприведённой схемы деления на отдельные функциональные части, Code_Aster является процессором (решателем). В Code_Aster представлены некоторые возможности, относящиеся к функциям препроцессора и постпроцессора, но они носят вспомогательный характер.

Основная часть расчётных модулей Code_Aster написана на алгоритмическом языке FORTRAN. Организация вычислительного процесса в целом возложена на часть Code_Aster, которая в технической документации по Code_Aster получила название супервайзер (supervisor). Супервайзер реализован на алгоритмическом языке Python.

Основная функция супервайзера Code_Aster заключается в управлении вычислительным процессом в соответствии с командами пользователя. Для реализации команд пользователя используется концепция управляющего языка, на котором пользователь определяет последовательность решения задачи. Список доступных команд Code_Aster хранится в каталоге команд. Последовательность команд пользователя оформляется в виде командного файла. Для реализации супервайзера, каталога команд и командного файла пользователя используется алгоритмический язык Python.

Командный файл пользователя представляет собой последовательность вызовов команд `Code_Aster`, определённых в каталоге команд. Команды пользователя могут иметь входные параметры, которые определяются ключевыми словами и значениями. Команды пользователя формируют выходные параметры, которые используются как входные параметры для последующих команд. При оформлении командного файла пользователь должен выполнять синтаксические правила языка Python и требования, которые определены в каталоге команд `Code_Aster`.

Выполнение задания пользователя начинается с синтаксического анализа командного файла. Этот анализ выполняет супервайзер `Code_Aster`. Если на этом этапе обнаруживается ошибка, то выдаётся сообщение об ошибке и выполнение задания прекращается.

На втором этапе супервайзер `Code_Aster` для каждой команды из командного файла создаёт объект `PythonETAPE`.

На третьем этапе супервайзер `Code_Aster` проверяет каждый объект `ETAPE` на соответствие требованиям каталога команд. Если требования не выполняются, то выдаётся сообщение об ошибке и выполнение задания пользователя прекращается.

На четвёртом этапе супервайзер `Code_Aster` организует выполнение команд. Для каждого объекта `ETAPE` организуется вызов соответствующего расчётного модуля, написанного на языке FORTRAN.

На начальном этапе супервайзер `Code_Aster` выполняет две задачи инициализации вычислительного процесса. Первая задача инициализации состоит в открытии стандартных файлов для ввода и вывода данных. Эти файлы используются для ввода команд пользователя и вывода информационных сообщений, сообщений об ошибках и результатов.

Вторая задача состоит в определении и открытии баз данных, которые представляют собой файлы прямого доступа. Управление базами данных осуществляется при помощи программного продукта JEVEUX.

Подготовленные пользователи могут использовать в командном файле все возможности языка Python. В частности, можно использовать циклы (`for`), проверки (`if`), исключения (`try, except`). Используя эти возможности, можно организовать адаптивные вычисления. Например, можно в цикле подбирать схему разбиения расчётной области на конечные элементы, используя некоторый критерий достижения требуемого результата. В командном файле `Code_Aster`, пользователь также может реализовать следующие задачи: чтение данных из файла или запись данных в файл; численные вычисления (например, используя `NumericalPython`); управление другими программами; использование графических приложений (`grace, gnuplot`).

Основное назначение программного комплекса `Code_Aster` заключается в анализе механического состояния сложных сооружений, конструкций и деталей. Первоначально `Code_Aster` разрабатывался для исследования состояния сооружений, конструкций и оборудования, используемого в ядерной энергетике. Поэтому большое внимание уделялось моделированию условий, в которых

функционируют такие объекты. В частности, рассматривались взаимодействие деформируемых конструкций с жидкостью, влияние температуры на механические свойства металлов, изменение свойств бетона с течением времени. В настоящее время область применения Code_Aster значительно расширилась. На данный момент Code_Aster можно рассматривать как процессор (решатель) общего назначения для анализа напряжённо-деформированного состояния сложных твердотельных систем.

Краткую характеристику текущего состояния Code_Aster можно представить следующим образом.

Общий объём программного кода, реализующего Code_Aster, превысил 1000000 строк.

Количество команд пользователя, хранящихся в каталоге команд и распознаваемых супервайзером Code_Aster, превысило 200 команд.

На данный момент можно моделировать и решать следующие классы задач:

- статические и квазистатические задачи в линейной и нелинейной постановках;
- динамические задачи в линейной и нелинейной постановках;
- задачи, связанные с разрушением и усталостью металлов;
- задачи взаимодействия сооружения с основанием и сооружения с жидкостью.

Реализованы следующие типы анализа:

- стандартный;
- метод разложения Фурье;
- метод суперэлементов;
- модальное разложение;
- адаптивные сетки конечных элементов.

В нелинейных постановках статических и динамических задач можно учитывать следующие явления:

- большие деформации и большие перемещения;
- следящие внешние силы;
- контакт и трение;
- потеря устойчивости.

На сегодняшний момент в программном комплексе Code_Aster реализовано более 100 моделей поведения материалов. Можно отметить следующие классы моделей:

- нелинейная упругость;
- гиперупругость;
- вязкоупругость;
- пластичность;
- вязкопластичность;
- повреждения;
- металлургические явления;
- гидратация и ползучесть бетона.

Программный комплекс Code_Aster представляет широкие возможности по решению задач динамики. Можно отметить следующие основные возможности:

- определение форм и частот собственных колебаний (с учётом или без учёта демпфирования);
- решение нестационарных задач динамики (прямыми методами, методом модального разложения, методом подструктур);
- решение нелинейных задач динамики (пластичность, повреждение, контакт, трение).

Библиотека программного комплекса Code_Aster включает более 400 различных типов конечных элементов. Среди основных типов конечных элементов можно отметить:

- плоские конечные элементы;
- осесимметричные конечные элементы;
- объёмные конечные элементы;
- стержневые конечные элементы для расчёта ферм;
- стержневые конечные элементы для расчёта рам;
- конечные элементы для расчёта пластин;
- конечные элементы для расчёта мембран;
- конечные элементы для расчёта оболочек;
- конечные элементы для расчёта гибких нитей;
- недеформируемые конечные элементы.

Возможны различные варианты использования программного комплекса Code_Aster для решения задач.

Первый вариант использования заключается в использовании только Code_Aster и любого текстового редактора, сохраняющего текст в виде ASCII файла. Схематично первый вариант использования Code_Aster показан на рисунке 2. На первом этапе пользователь посредством текстового редактора создаёт два файла.

Первый файл содержит данные о геометрии модели. В частности, этот файл должен определять следующие сущности:

- узлы — точки, определяемые именами и их координатами в двумерном или трёхмерном пространстве;
- элементы сетки, которые представляют собой именованные линейные, плоские или объёмные фигуры;
- именованные группы узлов;
- именованные группы элементов сетки.

Группы узлов и элементов сетки используются для упрощения в дальнейшем определения характеристик конечно элементной модели и анализа результатов. Имена всем перечисленным сущностям назначает пользователь.

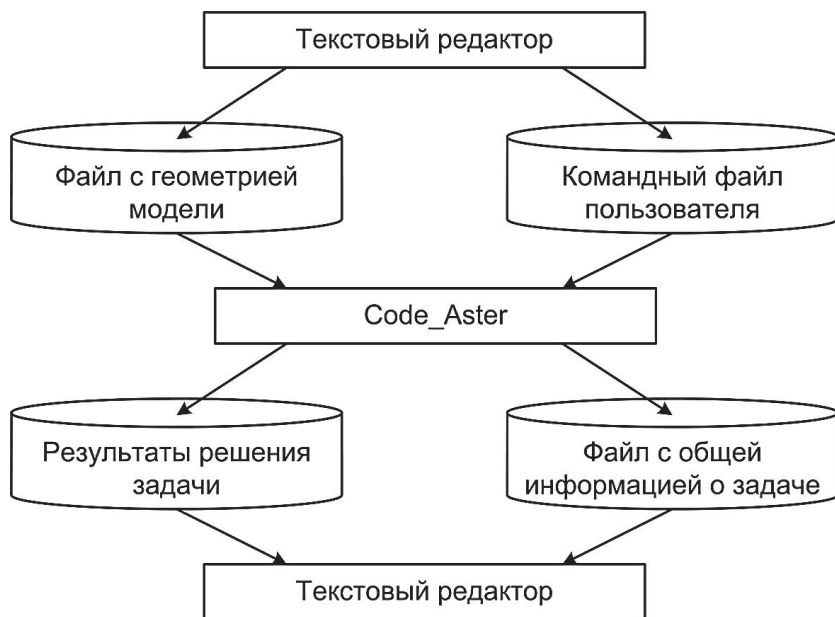


Рис. 2. Первый вариант использования Code_Aster

Второй файл содержит команды пользователя, представляет собой программу на языке Python и выполняет следующие задачи:

- указывает местоположение и формат файла с данными о геометрии модели;
- определяет, какие типы конечных элементов необходимо использовать для элементов сетки;
- определяет характеристики выбранных конечных элементов;
- указывает, какие операции необходимо выполнить в процессе решения задачи;
- определяет, какие результаты, где и в каком формате необходимо сохранить.

На втором этапе пользователь запускает Code_Aster и указывает местоположение командного файла. Code_Aster анализирует содержание командного файла и выполняет команды, если они не содержат ошибок. В результате Code_Aster формирует как минимум два ASCII файла. В одном из них содержится общая информация о решаемой задаче, информация об использованном процессорном времени, информация об использованной оперативной и внешней памяти, предупреждения и сообщения об ошибках. Второй файл содержит результаты расчёта, которые пользователь затребовал в командном файле.

На третьем этапе пользователь может просмотреть содержимое двух файлов, сформированных Code_Aster, с помощью любого текстового редактора.

Вышеописанный вариант носит скорее академический характер и пригоден для решения простейших тестовых задач. Геометрия моделей в реальных задачах может быть очень сложной. Подготовка файлов с геометрией таких моделей требует использования специальных программных продуктов, называе-

мых препроцессорами. Можно отметить следующие программные продукты, которые работают под операционной системой Linux и могут быть использованы в качестве препроцессора совместно с Code_Aster: IDEAS, GIBI, GMSH. Вариант использования Code_Aster совместно со свободно распространяемым продуктом GMSH показан на рисунке 3.

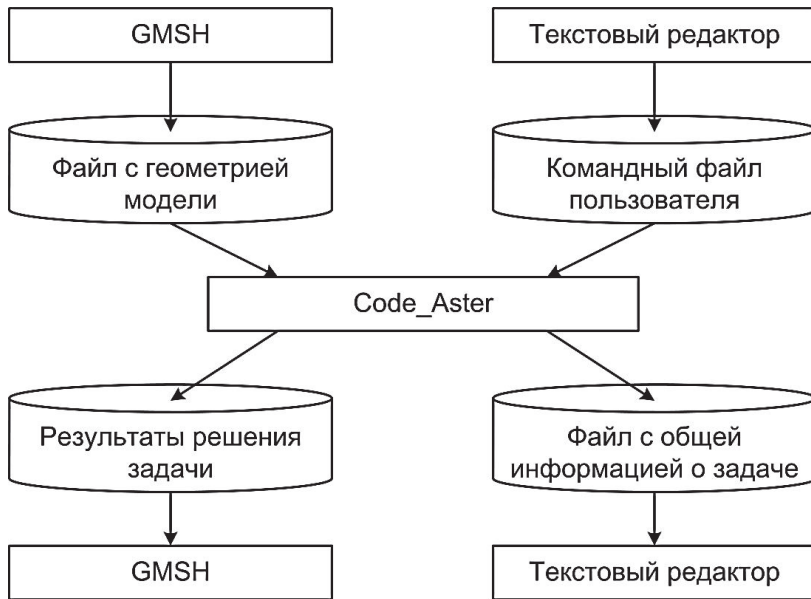


Рис. 3. Второй вариант использования Code_Aster

Третий вариант использования Code_Aster опирается на свободно распространяемый продукт Salome, разработанный фирмой «EDF R&D», и программный продукт Salome-Меса, разработанный той же фирмой. Программный продукт Salome используется в качестве препроцессора и постпроцессора. Программный продукт Salome-Меса используется в качестве платформы, которая обеспечивает взаимодействие между Salome и Code_Aster, а также формирование командного файла. Третий вариант использования Code_Aster показан на рисунке 4.

Второй вариант использования Code_Aster, показанный на рисунке 3, требует от пользователя знания синтаксиса команд Code_Aster и алгоритмического языка Python. Взамен пользователь может использовать все возможности языка Python и программного комплекса Code_Aster.

Третий вариант использования Code_Aster не требует от пользователя знания синтаксиса языка Python и командного языка Code_Aster, однако позволяет решать лишь ограниченный набор типовых задач. Доступ ко всем возможностям Code_Aster можно получить посредством ручного редактирования командного файла. Ручное редактирование командного файла возможно при помощи текстового редактора, запускаемого из окружения Salome-Меса.

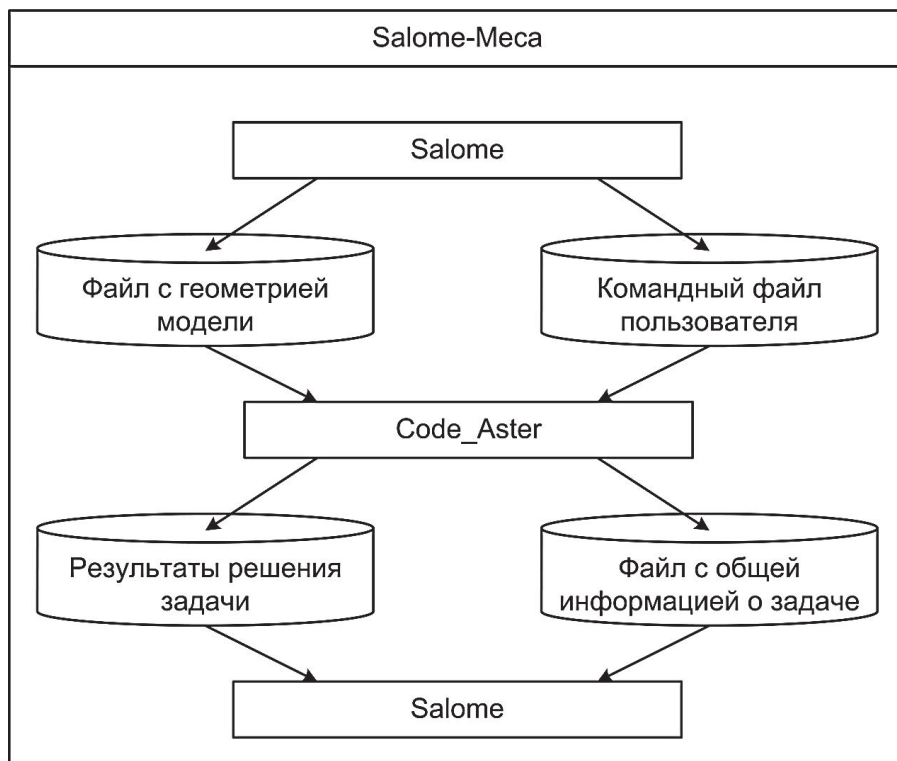


Рис. 4. Третий вариант использования Code_Aster

В заключение можно отметить, что программный комплекс Code_Aster представляет большой интерес с точки зрения использования в учебном процессе при изучении строительной механики и теории упругости. Широкие возможности по решению нелинейных задач статики и динамики позволяют использовать этот программный комплекс при подготовке бакалавров, магистров и аспирантов. На первом этапе необходимо разработать методическое пособие для студентов и преподавателей, описывающее основы использования программного комплекса Code_Aster при решении типовых задач строительной механики и теории упругости. На втором этапе целесообразно использовать программный комплекс Code_Aster в рамках научно-исследовательской работы студентов. Третий этап предусматривает использование этого программного комплекса на лабораторных работах по строительной механике и теории упругости.

Литература

1. Перельмутер А. В., Сливкер В. И. Расчётные модели сооружений и возможность их анализа. М.: ДМК Пресс, 2007. 600 с.
2. Сеницын С. Б., Строительная механика в методе конечных элементов стержневых систем. М.: Издательство АСВ, 2002. 320 с.
3. Hutton D. V. Fundamentals of finite element analysis. New York: McGraw-Hill, 2004. 494 p.

4. Zienkiewicz O. C. The finite element method in engineering science. London: McGraw-Hill, 1971. 521 p.
5. Zienkiewicz O. C., Taylor R. L. The finite element method. Vol. 1. Basic formulation and linear problems. London, New York: McGraw-Hill, 1989. 648 p.
6. Zienkiewicz O. C., Taylor R. L. The finite element method. Vol. 2. Solid and fluid mechanics. London, New York: McGraw-Hill, 1991. 807 p.
7. Zienkiewicz O. C., Taylor R. L. The finite element method. Vol. 1. The basis. Oxford: Butterworth-Heinemann, 2000. 707 p.

P. I. Safronov

USAGE OF SOFTWARE CODE_ASTER FOR SOLVING TASKS OF STRUCTURAL MECHANICS AND THEORY OF ELASTICITY

The basic features of the software Code_Aster are considered. Software Code_Aster is used for calculate the complex mechanical models by the finite element method. The possibility of using the software Code_Aster for the study subjects «Structural Mechanics» and «Theory of Elasticity» is considered.

Keywords: finite element method, software Code_Aster, structural mechanics, theory of elasticity.

Сафронов Павел Иванович — доцент кафедры «Строительная механика» ФГБОУ ВПО ПсковГУ, канд. техн. наук, safronov.pskgu@gmail.com.