

**Основы LS-DYNA.
Диаграмма предельных деформаций (FLD).
Создание и верификация моделей поведения материалов.**

Методическое пособие

(практическая часть научно-исследовательской работы студентов при кафедре
Динамики и прочности машин
Пермского национального исследовательского политехнического университета)

Составитель: Казанцев А.В.
e-mail: alexkazancev@bk.ru

Версия 1.1 от 08.04.2018

Публичная лицензия GNU FDL v1.3

Пермь, 2017-2018

Copyright © Kazantsev A.V. (Казанцев А.В.)

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with the Invariant Sections being "FORTRAN Sources of UMAT" ("Исходные тексты UM на языке FORTRAN"). See a text of the License at <https://www.gnu.org/licenses/fdl.html>.

Оглавление

Часть первая

1. Вводные сведения.....	6
1.1. Области применения и возможности.....	6
1.2. Интегрирование уравнения движения в матричной форме.....	8
1.3. Система единиц измерений.....	9
2. Структура и ключевые слова расчётного файла LS-DYNA.....	10
2.2. Основные файлы, создаваемые солвером во время расчёта.....	11
2.3. Тип расчётного файла (первая строка k-файла).....	11
2.4. Заголовок, строки комментариев, окончание расчётного файла.....	11
2.5. Синтаксис ключевых слов.....	11
2.6. Задание переменной.....	12
2.7. Массив узловых координат.....	12
2.8. Элемент и его характеристики (секция).....	13
1) Твёрдотельный элемент.....	13
2) Оболочка.....	13
3) Балка.....	13
4) Пружинка или демпфер.....	13
2.9. Массив элементов.....	14
1) 8-ми узловой твёрдотельный элемент.....	14
2) Оболочка.....	14
3) Оболочка с непосредственным указанием толщины.....	14
4) Балка.....	14
5) Сосредоточенная масса.....	14
6) Дискретный элемент между двумя узлами (пружинка или демпфер).....	14
2.10. Задание «части» (part, деталь).....	15
1) Каждая часть описывается своей структурой вида.....	15
2) Все части задаются одним ключевым словом *PART.....	15
2.11. Группы (set, совокупности).....	16
1) Группа узлов, списком.....	16
2) Группа узлов, в столбец.....	16
3) Группа оболочек.....	16
4) Группа частей.....	16
5) Сегмент (область из треугольников или четырёхугольников).....	16
2.12. Табличная функция (интерполяционная кривая по точкам).....	17
2.13. Материал.....	17
1) Упругий изотропный.....	17
2) Недеформируемый (rigid).....	17
3) Изотропный упруго-пластический.....	17
4) Пластика с кинематическим упрочнением.....	17
5) Пружинка.....	18
6) Вязкий элемент (демпфер).....	18
7) Трансверсально анизотропный упруго-пластический (№ 37).....	18
2.14. Нагрузки.....	18
1) Приложить в центре масс жёсткого тела сосредоточенную силу или момент.....	18
2) Нагрузка, прикладываемая к набору узлов.....	18
3) Нагрузка, прикладываемая к набору оболочечных элементов.....	18
4) Сосредоточенная сила в узле, или в наборе узлов (см. *LOAD_NODE_SET).....	18
5) Давление.....	19
6) Нагрузка на сегмент (давление).....	19
7) Ускорение поступательное или угловое, прикладываемое к телу.....	19

8) Ускорение свободного падения.....	19
9) Граничные условия, перемещение части.....	19
10) Переменное во времени ГУ для узла.....	20
11) Переменное во времени ГУ для узлов.....	20
12) Переменное во времени ГУ для узлового набора.....	20
13) Фиксация степеней свободы узла.....	20
14) Фиксация степеней свободы узлов.....	20
15) Начальная скорость в указанных узлах.....	20
16) Начальная скорость в узловом наборе.....	20
17) Начальная скорость узлов в указанной области (параллелепипеде).....	21
18) Начальная скорость части или набора частей.....	21
2.15. Связывание степеней свободы.....	21
1) Шарниры.....	21
2) Жёстко связать узловой набор 1 с частью 1.....	21
3) Связать указанные степени свободы выбранных узлов.....	21
4) Связать («склеить») узел оболочки с узлом твёрдотельника.....	22
5) Склеить совпадающие по координатам узлы с возможностью разрушения связи.....	22
2.16. Контакты.....	22
1) Универсальный.....	22
2) Прочие типы контактов, включая автоматические.....	22
2.17. Управляющие параметры.....	23
1) Корректировка шага по времени в явном анализе.....	23
2) Время моделирования процесса.....	23
3) Активация адаптивного перестроения сетки.....	23
4) Объёмная вязкость.....	23
5) Конструкционное демпфирование.....	23
6) Диссипация энергии.....	23
7) Неустойчивость формы конечного элемента по типу «песочных часов».....	24
2.18. База данных результатов.....	24
1) Шаг по времени для записи результатов на диск.....	24
2) Шаг записи результатов для предварительно выбранных элементов, узлов, частей.....	24
3) История выбранных узлов, оболочек, твёрдотельников, шарниров.....	24
4) Группа узловых сил.....	24
5) Формат сохраняемой базы данных.....	25
6) Шаг выдачи глобальной статистики в лог-файл.....	25
7) Шаг выдачи суммарной энергии системы.....	25
2.19. Дополнительные ключевые слова.....	25
1) Переключить анализ на неявный.....	25
2) Указать, что выбранные части станут недеформируемыми.....	25
3) Включить ключевые слова из текстового файла с именем FILE1.....	25
4) Время останова, заменяет значение, указанное в карте *CONTROL_TERMINATION.....	26
5) Вектор (несколько векторов) по координатам двух точек.....	26
6) Сохранять в базе данных результатов переменные истории.....	26
2.20. Принудительный останов решателя, рестарт решателя.....	26

Часть вторая

3. Диаграмма предельных деформаций (FLD).....	28
3.1. Моделирование листовой формовки.....	28
3.2. Диаграмма предельных деформаций Forming-limit diagram (FLD).....	30
3.3. Области FLD.....	31
3.4. Оценка предельной деформации в LS-DYNA.....	33
3.5. Анизотропия листового металла.....	35
3.6. Задание: построить FLD по данным вычислительного эксперимента.....	36

Часть третья

4. Создание нового материала в LS-DYNA.....	39
4.1. Введение.....	39
4.2. Этапы сборки бинарника с новым материалом (единый файл).....	40
4.3. Примеры скриптов автосборки (Makefile).....	44
1) Для компилятора Intel Fortran, среда Intel XE 2013 (Linux).....	44
2) Для компилятора gfortran (Linux).....	44
3) Сборка динамической библиотеки материалов (Linux) [71].....	45
4) Линковка стороннего объектника [32].....	45
4.4. Дополнения к коду решателя. User Subroutines.....	46
4.5. Новый материал. User Material (UM, UMAT).....	47
4.6. Базовый синтаксис FORTRAN и пример UM.....	48
4.7. Подключение UM в расчётном файле.....	50
4.8. Массивы и переменные в UM. Баги, недокументированные особенности, информация к сведению.....	50
5. Примеры UM.....	53
5.1. Линейно упругий материал.....	53
5.2. Изотропный упруго-пластический материал.....	54
5.3. Полимер для термомеханических задач.....	56
6. Верификация UM.....	58
6.1. Верификация линейно-упругого материала.....	58
6.2. Верификация упруго-пластического материала с упрочнением.....	59
Литература, ссылки на примеры и видеоролики по основам LS-DYNA.....	60
Источники по UM и FLD.....	60
Литература по вычислительной механике.....	63
7. Исходные тексты UM на языке FORTRAN.....	64
7.1. Линейно-упругий материал.....	64
7.2. Линейный ортотропный материал (векторная подпрограмма).....	67
7.3. Изотропный билинейный материал с уравнением состояния.....	70
7.4. Резина (применение деформационного градиента).....	71
7.5. Изотропный упругий для тепловых задач.....	76
7.6. Упруго-пластический материал по Мизесу и алгоритм интегрирования напряжений.....	79
7.7. Полимер для термомеханических задач.....	84
7.8. Линейный упругий материал для неявного анализа.....	86
7.9. Упруго-пластический материал с изотропным упрочнением.....	87
7.10. Композит с разрушением.....	88
7.11. Модель анизотропной пластичности Хилла.....	90
8. Примеры расчётных файлов.....	93
8.1. Пружинный маятник с демпфером.....	93
8.2. Плоский математический маятник.....	94
8.3. Падение кубика на пластинку.....	95
8.4. Крутильный маятник.....	96

Часть первая

И будет очень обидно, если они лягут здесь и заснут... хотя они могли дойти. Это будет ужасно обидно. И поэтому надо.

Аркадий и Борис Стругацкие. Страна багровых туч.

1. Вводные сведения

1.1. Области применения и возможности

В прикладных исследованиях быстропротекающих процессов, а именно, ударов, взрывов, разрушений, программный комплекс LS-DYNA¹ применяется как явный (и неявный) решатель для нелинейных задач МДТТ, МЖГ, теплопереноса, а также связанных задач. Решатель учитывает силы инерции и находит ускорения, по ним - скорости и перемещения элементов конечно-элементной сетки, напряжения и деформации в элементах, энергетические параметры, усилия и моменты, нормальные и касательные силы, контактные параметры; все искомые величины зависят от функции времени.

Ряд прикладных задач, решаемых в LS-DYNA:

- оценка сопротивляемости удару – пассивная безопасность: автомобили, летательные аппараты, поезда, суда;
- анализ динамической прочности автомобильных (и др.) комплектующих: кузова, бамперов, колёсных дисков, рулевых колонок и т.д.;
- оценка безопасности пассажира: взаимодействие модели человека с воздушной подушкой и ремнём безопасности, прорыв подушки безопасности и др.;
- отрыв лопаток турбинных двигателей;
- птицестойкость;
- взаимодействие потоков жидкости и газа с конструкцией;
- взрывная нагрузка на изделия;
- задачи проникания (пробивание броневой пластины, внедрение в грунт свай и т.п.);
- анализ ячеистых, сотовых и тонкостенных кессонных конструкций, жестяных контейнеров;
- расчёт сварных, заклёпочных и болтовых соединений.

Доступные расчётные схемы:

- явная;
- неявная (прямой решатель с учётом разрежённости матриц, итерационный);
- бессеточные: метод выглаженных частиц (Smoothed Particle Hydrodynamics – SPH), метод Галёркина (Element Free Galerkin method – EFG);
- и др. (см. руководства).

Встроены процедуры автоматической/ручной адаптивной перестройки КЭ-сетки при вырождении элементов и сглаживания КЭ-сетки, эффективные алгоритмы решения контактных задач.

Порядка 50-ти моделей контактно-ударных алгоритмов учитывают особенности взаимодействия контактирующих сред: скольжение, сухое и вязкое трение, теплопередача,

1 Собственность Livermore Software Technology Corporation (LSTC)

отлипание, вязкое и хрупкое разрушение, пробивание, контакт абсолютно твёрдых тел с деформируемыми структурами и между собой, сварные, болтовые и заклёпочные соединения.

Большая библиотека элементов включает помимо балочных, оболочечных и объёмных элементов и узкоспециализированные элементы, например элемент ремня безопасности в автомобиле, а также большой набор дискретных элементов. Кроме элементов с полной схемой интегрирования доступны быстро интегрируемые элементы с упрощенной (одноточечной) схемой. Все элементы векторизованы для параллельных вычислений.

Решатель располагает более чем двумястами встроенными моделями поведения материалов: металлы, композиты, грунты, керамика, стёкла, пенополиуретаны, жидкие и газообразные среды, взрывчатые вещества в условиях термомеханического воздействия, в условиях больших давлений, скоростей деформаций и температур. Многие модели поведения допускают разрушение по какому-либо критерию разрушения.

Наиболее известные встроенные модели материалов:

- абсолютно жёсткие тела;
- упругих: изотропная; ортотропная; анизотропная; термоупругая; упругая с разрушением; вязкоупругая;
- упруговязкопластических: упругопластические с изотропным, кинематическим и комбинированным упрочнением, в том числе с зависимостью свойств от температуры и скорости деформации; упругопластические с анизотропным упрочнением; упругопластическая модель Стенберга-Гунана и Джонсона-Кука; полной пластичности; сверхпластичности;
- пены: гиперупругая; разрушаемая и не разрушаемая; изотропная; целлюлозы; уретановая пена с гистерезисом;
- эластомеры и резины: Муни-Ривлина, Блейца-Ко, Огдена, Арруда-Бойса;
- линейная вязкоупругая модель;
- ткани;
- стёкла;
- древесина;
- акустическая среда;
- композиты, включая модели слоистых, матричных и тканевых, с разрушением и без;
- грунты, геологические породы;
- кевлар с повреждениями;
- биомеханические материалы, включая ткани сердца;
- специальные модели для задач гидродинамики;
- модели, которые могут быть созданы пользователем.

В области **обработки металлов давлением LS-DYNA** применяется для анализа процесса штамповки листового материала с конца 80-х годов. Возможности применения в ОМД: моделирование процессов прокатки профилей, листовой штамповки, профилирования в роликах, учёт упругого пружинения (springback), моделирование протяжки и глубокой вытяжки, волочения, листовая и трубная гидроформовка с большими деформациями, суперпластическая формовка, резка, литьё, многоступенчатые процессы, оценка формирующего процесса, проектирование инструмента и др. Аналогичные задачи могут решаться для стёкол и пластика.

Для моделирования процессов листовой формовки разработан специальный класс контактов, их применение понижает итоговую вычислительную сложность задачи (см. лит. в главе 5 [31]).

В LS-DYNA и в других кодах, достоверное моделирование поведения листового металла в процессе формовки стала возможным после внедрения ряда алгоритмов, к ним относят [53]:

- адаптивное перестроение сетки заготовки; генерация специализированной сетки для контактных задач формовки;
- модели поведения материала для листовой штамповки и формовки, учитывающие анизотропию листа, такие как Barlat (MAT_036) и Hill (MAT_037);
- развитие решателя: распараллеливание явной схемы; неявные схемы для учёта ускорения свободного падения и пружинения; одношаговые методы для оценки формы заготовки;

- расширенная постобработка для процессов листовой формовки: построение FLD, линий скольжения, нанесение сетки окружностей и т.д.

LS-DYNA поставляется под основные аппаратно-программные платформы: Cray, Intel Itanium, IBM Power, Apple macOS, SGI, Microsoft Windows, Linux; решатель векторизован, распараллелен для систем с общей (SMP) и распределённой (MPP) памятью.

Присутствуют возможности перепрограммирования решателя.

Основной пре/постпроцессор: LS-PREPOST, свободно распространяемый.

1.2. Интегрирование уравнения движения в матричной форме

В **явной** схеме с центральными разностями уравнение движения разрешается по ускорению:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \sum \{F\}; \quad (1)$$

$$\{\ddot{u}_t\} = [M]^{-1} (\{F_t^{\text{внешн}}\} - \{F_t^{\text{внутр}}\}); \quad (2)$$

$$\{\dot{u}_{t+\Delta t/2}\} = \{\dot{u}_{t-\Delta t/2}\} + \{\ddot{u}_t\} \Delta t; \quad (3)$$

$$\{u_{t+\Delta t}\} = \{u_t\} + \{\dot{u}_{t+\Delta t/2}\} \Delta t_{t+\Delta t/2}; \quad (4)$$

$$\Delta t_{t\pm\Delta t/2} = \frac{\Delta t_t \pm \Delta t_{t+\Delta t}}{2}. \quad (5)$$

Матрица масс $[M]$ сводится к диагональной (сосредоточенной)², тогда обращение в (2) становится тривиальным:

$$[M] = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \dots & \dots & \dots & \dots \\ m_{n1} & m_{n2} & \dots & m_{nn} \end{bmatrix} \rightarrow [M] \approx [M^L] = \begin{bmatrix} m_{11}^L & 0 & \dots & 0 \\ 0 & m_{22}^L & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & m_{nn}^L \end{bmatrix}; \quad (6)$$

$$\rightarrow [M]^{-1} \approx [M^L]^{-1} = \begin{bmatrix} \frac{1}{m_{11}^L} & 0 & \dots & 0 \\ 0 & \frac{1}{m_{22}^L} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{m_{nn}^L} \end{bmatrix}; \quad (7)$$

- уравнения становятся разьединёнными и могут быть решены явно, проверка сходимости не требуется;
- все нелинейности (включая контакт) включены в вектор сил;

2 Подробнее Zhu, J., Z. R. L. Taylor, and O. C. Zienkiewicz. "The finite element method: its basis and fundamentals." (2005): 54-102, section 16.2.4; Felippa, Carlos A., Qiong Guo, and K. C. Park. "Mass matrix templates: General description and 1d examples." Archives of Computational Methods in Engineering 22.1 (2015): 1-65.

- единственный критерий устойчивости решения - очень малый временной шаг, который зависит от скорости звука в выбранном материале, и согласно критерию Куранта (CFL)³:

$$\Delta t \leq \frac{l}{c}, \quad (8)$$

где l – линейный размер наименьшего КЭ, c – скорость звука. Для стабильности правая часть (8) домножается на некоторый коэффициент, меньший 1.

В **неявной схеме** применяется одношаговое прямое интегрирование по Ньюмарку с заданным шагом Δt_t . Шаг по времени не обязательно фиксирован, матрица масс может сводиться к диагональной. Оптимальным считается использование 10 и более временных шагов на период отклика для максимальной частоты рассматриваемой конструкции.

Приведём этапы схемы Ньюмарка. Алгоритм начинается с вычисления постоянных:

$$\alpha_1 = \frac{2}{\Delta t_{t+\Delta t}}; \alpha_2 = \alpha_1^2; \alpha_3 = 2\alpha_1; \alpha_4 = \frac{\Delta t_{t+\Delta t}}{2}. \quad (9)$$

Далее вычисляется динамическая матрица жёсткости

$$[K_D] = [K] + \alpha_1 [C] + \alpha_2 [M]; \quad (10)$$

вспомогательные векторы

$$\{V_1\} = \alpha_3 \{\dot{u}_t\} + \{\ddot{u}_t\}; \{V_2\} = \alpha_1 \{u_t\} + \{\dot{u}_t\}; \{V_3\} = \alpha_2 \{u_t\} + \{V_1\}; \quad (11)$$

динамический вектор внешних сил

$$\{F_D\} = \{F_{t+\Delta t}\} + [C]\{V_2\} + [M]\{V_3\}. \quad (12)$$

Решается СЛАУ

$$[K_D]\{u_{t+\Delta t}\} = \{F_D\}. \quad (13)$$

Вычисляются векторы ускорений и скоростей для следующего шага интегрирования:

$$\begin{aligned} \{\ddot{u}_{t+\Delta t}\} &= \alpha_2 (\{u_{t+\Delta t}\} - \{u_t\}) - \{V_1\}; \\ \{\dot{u}_{t+\Delta t}\} &= \{\dot{u}_t\} + \alpha_4 (\{\ddot{u}_t\} + \{\ddot{u}_{t+\Delta t}\}); \end{aligned} \quad (14)$$

и алгоритм замыкается на шаг (11), подробнее [73, 74].

1.3. Система единиц измерений

Как и в ANSYS⁴, в LS-DYNA SEI определяет пользователь, см. табл. 1. Размерность каждой единицы согласовывается по правилу: сила=масса*ускорение, ускорение=длина/время².

Таблица 1. Распространённые SEI (согласно [1])

	масса	длина	время	сила	напряжение, давление	плотность	модуль Юнга	ускорение свободного падения
СИ	кг	м	с	Н	Па	7.83e3	2.1e11	9.8
ГСМ	г	см	мкс	1e7	Мбар	7.83	2.1	9.8e-10
ОМД	тонна	мм	с	Н	МПа	7.83e-9	2.1e5	9.8e3

Система ОМД наиболее удобна в процессах обработки металлов давлением. Система ГСМ применяется в процессах с длительностью порядка микросекунд и миллисекунд, например, в процессах взрыва. Система СИ общеизвестна. Иные варианты см. в Keyword Manual.

3 https://en.wikipedia.org/wiki/Courant-Friedrichs-Lewy_condition

4 Собственность корпорации ANSYS Inc.

2. Структура и ключевые слова расчётного файла LS-DYNA

2.1. Таблица часто используемых ключевых слов

Таблица 2. Широко применяемые ключевые слова

Группа	Объект	Ключевое слово
Сетка конечных элементов	Узел	*NODE
	Элемент	*ELEMENT_BEAM, *ELEMENT_SHELL *ELEMENT_SOLID, *ELEMENT_TSHELL
	Дискретный элемент	*ELEMENT_DISCRETE, *ELEMENT_MASS *ELEMENT_SEATBELT_option
Материал	Часть (с заданным материалом, типом элемента, уравнением состояния, контролем потери формы и пр.)	*PART
	Материал	*MAT_option
	Характеристика элемента	*SECTION_BEAM, *SECTION_SHELL *SECTION_SOLID, *SECTION_TSHELL
	Характеристики дискретного элемента	*SECTION_DISCRETE, *SECTION_SELTBELT
	Уравнение состояния	*EOS_option
	Контроль потери формы	*CONTROL_HOURLASS, *HOURLASS
Контакт с жёсткой стенкой	Свойства контакта по-умолчанию	*CONTROL_CONTACT
	Задание контакта	*CONTACT_option
	Задание жёсткой стенки	*RIGIDWALL_option
	Связывание степеней свободы	*NODE, *BOUNDARY_SPC_OPTION
Граничные условия и нагрузки	Ускорение свободного падения	*LOAD_BODY_option
	Узловая нагрузка	*LOAD_NODE
	Давление	*LOAD_SEGMEAL_option *LOAD_SHELL_option
	Тепловая нагрузка	*LOAD_THERMAL_option
	Табличная функция	*DEFINE_CURVE
Связи и шарниры	Связывание узлов	*CONSTRAINED_NODE_SET
	Сварка	*CONSTRAINED_GENERALIZED_WELD_option, *CONSTRAINED_SPOT_WELD
	Клёпка	*CONSTRAINED_RIVET
Управление базами данных результатов расчёта	Значения по умолчанию	*CONTROL_option
	Содержимое БД истории	*DATABASE_option
	Двоичные файлы графики, истории и файлы рестарта	*DATABASE_BINARY_option
	Выбор пунктов БД истории	*DATABASE_HISTORY_option
	Узловые силы и реакции	*DATABASE_NODAL_FORCE_GROUP
Останов решателя	Время останова	*CONTROL_TERMINATION
	Количество итераций до останова	*CONTROL_TERMINATION
	Время работы CPU	*CONTROL_CPU
	Предельное смещение или касание	*TERMINATION_NODE

2.2. Основные файлы, создаваемые солвером во время расчёта

Имена по умолчанию, подробнее см. LS-DYNA Keyword Manual и Главу 4 [32]:

- `messag` – короткий лог-файл, в основном выводит текущий временной шаг, текущее время в расчёте, а также ошибки, найденные в синтаксисе `*.k` – файла;
- `D3HSP` – полный лог-файл, содержит всю информацию о выбранных ключевых словах, узлах, элементах, материалах, деталях, энергетические характеристики модели на каждом расчётном шаге и прочее;
- `D3PLOT` – двоичный файл со всеми результатами для графического постпроцессинга;
- `D3THDT` – двоичный файл с результатами для выбранных элементов/узлов и т.д., предназначенный для графической постобработки;
- `D3DUMP` – двоичный файл рестарта анализа.

2.3. Тип расчётного файла (первая строка k-файла)

```
*KEYWORD <memory> NCPU=<n>
```

`*KEYWORD` – используется файл в формате ключевых слов,
`<memory>=80000000` – количество выделяемой памяти в словах (~ 4*80 МБайт),
`<n>=2` – количество ядер процессора.

2.4. Заголовок, строки комментариев, окончание расчётного файла

```
*Title  

<Название задачи транслитом>
```

\$ - символ начала строки комментария; примеры закомментированных строк:

```
$ с-ма единиц: тонна, мм, секунда, Ньютон, МПа, Н*мм  

$ part1: фиксированное абс.жёсткое тело  

$ part2: свинцовый шарик, начальная скорость 400 мм/с  

$ part3: оболочка, материал пластика с кинематич. упрочнением
```

Расчётный файл завершается ключевым словом:

```
*END
```

2.5. Синтаксис ключевых слов

На новой строке без пробела ставится символ звёздочки и пишется ключевое слово (через нижнее подчёркивание пишутся составные ключевые слова) в верхнем или нижнем регистре:

```
*<КЛЮЧЕВОЕ_СЛОВО>
```

Затем выбирается один из двух вариантов записи параметров ключевого слова.

1) Фиксированный формат - константы отделяются пробелами, под каждую константу отводится 10 символов, длина строки ограничивается 80-тью символами, длина целочисленных констант - не более 8-ми символов; константы желательно выравнивать по правому краю в указанных диапазонах:

```

$      2-10      12-20      22-30      32-40      42-50      52-60      62-70      72-80
$.>...>...1...>...2...>...3...>...4...>...5...>...6...>...7...>...8
      10      1E+01      1.      5.5 4.5000E+1

```

2) Свободный формат - запись параметров ключевого слова через запятую:

```

*<КЛЮЧЕВОЕ СЛОВО>
1255,0.0,45.,45.0,,
10, .05,,,,,1e-10, 50.0

```

Примечание: в расчётном файле ключевые слова могут располагаться в любом порядке.

Константы могут быть целочисленными (i - integer) и вещественными (f - float). Целая часть отделяется от дробной точкой. Показатель отделяется символом E. Попытка записи вещественного значения в переменную целого типа вызывает останов решателя.

2.6. Задание переменной

Значение переменной выравнивается справа на 20-й символ, длина имени переменной не более 8-ми символов:

```

*PARAMETER
I intgpar1      10
R realpar2      3.5
rRealpar3,4.0e5

```

Обращение к переменной производится с помощью знака амперсанд &:

```

*<КЛЮЧЕВОЕ СЛОВО>
1,0.12, 2e11, &realpar2, 0.5,,,,

```

2.7. Массив узловых координат

```

*NODE
$      nid      x      y      z      tc      rc
      3722 2.250000000E+00 2.500000000E+00 -1.25000000E+01
      231 1.200000E+02 0.000000E+00 2.000000E+00 7 7

```

nid – порядковый номер узла,

x, y, z – местоположение в ГСК,

tc, rc – константы, отвечающие за фиксацию степеней свободы узла, поступательные и вращательные соответственно, см. табл. 3.

Таблица 3. Значения параметров tc, rc

Значение	tc	rc
0	Узел неограничен в перемещениях	Узел неограничен в поворотах
1	Запрет перемещений по x	Запрет вращения вокруг x
2	Запрет перемещений по y	Запрет вращения вокруг y
3	Запрет перемещений по z	Запрет вращения вокруг z
4	Запрет перемещений по x и y	Запрет вращения вокруг x и y
5	Запрет перемещений по y и z	Запрет вращения вокруг y и z
6	Запрет перемещений по z и x	Запрет вращения вокруг z и x
7	Запрет перемещений по x, y, z	Запрет вращения вокруг x, y, z

2.8. Элемент и его характеристики (секция)

1) Твёрдотельный элемент

```
*SECTION_SOLID
$   sid   elform
$1,1 - порядковый номер 1, формулировка 1
$   - пустая строка указывает, что оставлены значения по умолчанию
      1      0
      2      0
```

sid – порядковый номер секции,

elform = 1 - секция определяет объёмный элемент с равномерным распределением напряжений (по умолчанию), прочие формулировки см. в Keyword Manual.

2) Оболочка

```
*SECTION_SHELL
$   sid   elform   shrf   nip   propt   qr/irid   icomp
      2      9      5      4      1      1      1
$   t1    t2    t3    t4    nloc
 0.015  0.015  0.015  0.015
```

elform – формулировка элемента, по умолчанию = 2 – Belytschko-Tsay, = 5 – мембрана Belytschko-Tsay, = 12 – ПНС (ХОУ), = 13 – ПДС (ХОУ), = 15 – осесимметричный твёрдотельный (y – ось симметрии), = 16 – полноценная интегрируемость, и т.д. (см. Keyword Manual),

shrf – коэффициент коррекции поперечного сдвига (= 5/6 для изотропного материала, подробнее см. Keyword Manual),

nip – число точек интегрирования по толщине, = 0.0 – 2 точки, = 1.0 – нет изгиба, 2.0-10. – от двух до десяти точек,

t1-t4 – толщина оболочки в каждом из узлов.

3) Балка

```
*SECTION_BEAM
$   sid   elform   shrf   qr/irid   cst
$   i     i     f     f     f
$   1     1     0.0
   45     3     1.00000   1.0
$ res:  a     iss     itt     irr     sa
$   f     f     f     f     f
$ 2.500E-01 2.500E-01 2.500E-01 2.500E-01 0.000E+00
   10.0
```

elform – формулировка, = 1 – по умолчанию (см. Keyword Manual), = 3 – стержень, = 6 – кабель, shrf – коэффициент сдвига, для прямоугольного сечения рекомендуется задавать = 5/6, по умолчанию = 1.0,

qr/irid – квадратуры Гаусса (см. Keyword Manual),

cst – вид поперечного сечения, = 0.0 – прямоугольное, 1.0 – круглое, 2.0 – заданное пользователем,

a – площадь сечения, iss, itt, irr – моменты инерции, см. Keyword Manual.

4) Пружинка или демпфер

```
*SECTION_DISCRETE
$ далее номер создаваемой секции
1
```

2.9. Массив элементов

1) 8-ми узловой твёрдотельный элемент

```
*ELEMENT_SOLID
$   eid      pid      n1      n2      n3      n4      n5      n6      n7      n8
    1         1         1         2         12        11        31        32        42        41
    37         2         91         92         95         94        100       101       104       103
    45         3        118        119       121        120       122       123       125       124
```

eid – порядковый номер элемента,
pid – порядковый № части (см. далее *part),
 n1-n8 – номера узлов.

2) Оболочка

```
*ELEMENT_SHELL
$   eid      pid      n1      n2      n3      n4
    6993         1     7547     7509     7148     7549
    3601         2     3722     3754     3755     3723
    4561         3     4714     4716     4717     4715
```

3) Оболочка с непосредственным указанием толщины

```
*ELEMENT_SHELL_THICKNESS
$   i      i      i      i      i      i
$   eid    pid    n1     n2     n3     n4
$           f           f           f           f
$      thic1      thic2      thic3      thic4
    1         1         1         10        11         2
    0.79000E+00    0.79000E+00    0.79000E+00    0.79000E+00
    2         1         10        19        20        11
    0.79000E+00    0.79000E+00    0.79000E+00    0.79000E+00
    3         1         19        28        29        20
    0.79000E+00    0.79000E+00    0.79000E+00    0.79000E+00
```

4) Балка

```
*ELEMENT_BEAM
$   eid      pid      n1      n2      n3
    45000         45     350     45004     45012
    45007         45     713     45011     45019
```

5) Сосредоточенная масса

```
*ELEMENT_MASS
$   eid      nid      mass
    65         65        1.000E-02
```

nid – номер узла,
mass – масса.

6) Дискретный элемент между двумя узлами (пружинка или демпфер)

```
*ELEMENT_DISCRETE
$   eid      pid      n1      n2      vid      s      pf      offset
    1         101         1        129         0    0.00000000E+00    1
    108        208       1505       1675         0    0.00000000E+00    1
```

n1, n2 – номера узлов,
 vid = 0 – пружинка/демпфер действуют вдоль линии, соединяющей узлы, если не 0 – указывается номер вектора (см. Keyword Manual).

2.10. Задание «части» (part, деталь)

Часть – это объект модели, на который нанесена сетка конечных элементов одного типа и которому присвоен один конкретный материал.⁵

1) Каждая часть описывается своей структурой вида

```
*PART
<Произвольное имя части>
$ далее идут константы
$      pid      sid      mid      eosid      hgid      (grav) adpopt
      1         1         1         0         0         0
```

pid – порядковый номер части,
 sid – тип элемента (см. *SECTION_),
 mid – номер материала (см. далее *MAT_),
 eosid – см. Keyword Manual,
 hgid – контроль формы, = 0 – используется значение по умолчанию,
 grav – см. Keyword Manual,
 adpopt – автоматическое перестроение сетки данной части в течение расчёта: < 0 – R-адаптация для 2D-твёрдотельников, = 0 – отключена, = 1 – H-адаптация для 3D-оболочек, = 2 – г-адаптация для осесимметричных 3D-твёрдотельников, 2D-твёрдотельников, 3D-тетраэдров, 3D-элементов бессеточного метода Галёркина, =3 – осесимметричная г-адаптация для 3D-твёрдотельников, = 9 – пассивная H-адаптация для 3D-оболочек.

Примечание: H-адаптация – каждый элемент делится на более мелкие с сохранением первоначальных узлов; R-адаптация – полное перестроение сетки (см. [1]).

Пример:

```
*PART
Прижим - жёсткий - зафиксирован по Z
$      pid      sid      mid      eosid      hgid      adpopt
      1         1         1         0         0         0
*PART
Заготовка - алюминий
$      pid      sid      mid      eosid      hgid      adpopt
      2         1         2         0         0         0
*PART
Матрица - жёсткий - зафиксирован
$      pid      sid      mid      eosid      hgid      adpopt
      3         2         3         0         0         0
```

2) Все части задаются одним ключевым словом *PART

```
*PART
$      pid      sid      mid      eosid      hgid      grav      adpopt
die-1
      1         1         1
nose-2
      2         1         2
tube-3
      3         2         3         0         0         0         1
```

⁵ Такое определение близко к определению, данному в ГОСТ 2.101-68, см. <https://ru.wikipedia.org/wiki/Деталь>

2.11. Группы (set, совокупности)

1) Группа узлов, списком

```
*SET_NODE_LIST
$   sid
   1
$   nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
   3279      3310      3341      3372      3403      3434      3465      3496
   3527      3558      3589      3620      3651
```

sid – порядковый номер,
nid1-nid8 – номера входящих в сборку узлов.

2) Группа узлов, в столбец

```
*SET_NODE_COLUMN
$   sid
   1
$   nid
   8901
   8904
```

3) Группа оболочек

```
*SET_SHELL_LIST
$   sid      da1      da2      da3      da4
   2
$   eid1     eid2     eid3     eid4     eid5     eid6     eid7     eid8
   221      222      223      224
```

eid1-eid8 – номера входящих в сборку элементов.

4) Группа частей

```
*SET_PART
$ в наборе одна единственная часть
$   sid
   1
$   pid
   1
*SET_PART_LIST
$ в наборе несколько частей
   1
   3       4       6
```

5) Сегмент (область из треугольников или четырёхугольников)

```
*SET_SEGMENT
$   sid
   1
$   n1      n2      n3      n4
   1       2       0       0
  185     186       0       0
```


2.12. Табличная функция (интерполяционная кривая по точкам)

```
*DEFINE_CURVE
$   lcid      sidr      scla      sclo      offa      offo
    1
$   abscissa (время)      ordinate (сила, давление, координата и т.д.)
    0.000E+00      0.000E+00
    3.200E-02      2.600E+01
    4.500E-02      6.000E-01
    8.000E-02      1.000E-01
```

lcid – порядковый номер созданной кривой,
 scla – коэффициент масштаба для оси абсцисс,
 sclo – -//- ординат.

2.13. Материал

1) Упругий изотропный

```
*MAT_ELASTIC
$   mid      ro      e      pr      da      db      k
    3  7.850E-04  3.000E+07  3.000E-01
$1,7850,2.1e11,0.3
```

mid – номер материала,
 ro – плотность,
 e – модуль Юнга,
 pr – коэффициент Пуассона.

2) Недеформируемый (rigid)

```
*MAT_RIGID
$   mid      ro      e      pr      n      couple      m      alias
    1  7.84e-4  30.00e+6  0.30
$   cmo      con1      con2
    1      7      7
$   lco/a1      a2      a3      v1      v2      v3
$
```

3) Изотропный упруго-пластический

```
*MAT_ISOTROPIC_ELASTIC_PLASTIC
$   i      f      f      f      f      f
$   mid      ro      g      sigy      etan      bulk
    17.3460e-04  1.133e+07  1.900e+05  0.000e+00  2.400e+07
```

4) Пластика с кинематическим упрочнением

```
*MAT_PLASTIC_KINEMATIC
$ Aluminum
$   mid      ro      e      pr      sigy      etan      beta
    1  2.70e-6  68.9  0.330  0.286  0.00689
$   src      srp      fs
$
```

5) Пружинка

```
*MAT_SPRING_ELASTIC
$1, &jestkost
      3      1
```

6) Вязкий элемент (демпфер)

```
*MAT_DAMPER_VISCOUS
$2, &dempfir
      143 2.300E+00
```

7) Трансверсально анизотропный упруго-пластический (№ 37)

```
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC
$      mid      ro      e      pr      sigy      etan      r      hlcid
      1 0.787E-08 0.207E+06      0.280      127.6      0.0      1.0      2.0
```

2.14. Нагрузки

1) Приложить в центре масс жёсткого тела сосредоточенную силу или момент

```
*LOAD_RIGID_BODY
$      pid      dof      lcid      sf      cid      m1      m2      m3
      2      2      2      -1.0
```

pid – номер компонента,

dof – степень свободы (= 1 - FX, = 2 - FY, = 3 - FZ, = 4 - MX, = 5 - MY, = 6 - MZ),

lcid – порядковый номер кривой или функции, которая задаёт силовой фактор во времени,

sf – масштабный коэффициент (отрицательный – действие в противоположном направлении), подробнее см. Keyword Manual.

2) Нагрузка, прикладываемая к набору узлов

```
*LOAD_NODE_SET
$      nsid      dof      lcid      sf      cid      m1      m2      m3
      2      3      1      0.5
$ Load node 46 in the negative z-direction.
$      2      3      1      -1.0
```

nsid – номер узловой сборки (см. *SET_NODE_LIST), остальные параметры аналогичны *LOAD_RIGID_BODY.

3) Нагрузка, прикладываемая к набору оболочечных элементов

```
*LOAD_SHELL_SET
$      i      i      f      f
$      esid      lcid      sf      at
      1      1 3.000E+02
```

esid – тип элемента или номер набора оболочек (см. *ELEMENT_SHELL SID и *SET_SHELL SSID).

4) Сосредоточенная сила в узле, или в наборе узлов (см. *LOAD_NODE_SET)

```
*LOAD_NODE_POINT
$      nid      dof      lcid      sf      cid      m1      m2      m3
```

91	3	2-1.000E+00	0
92	3	2-1.000E+00	0
93	3	2-1.000E+00	0

5) Давление

```
*LOAD_SHELL_ELEMENT
$   esid      lcid      sf      at
    30001      5 -1.00E+00    0.0
    30448      5 -1.00E+00    0.0
```

esid – порядковый номер элемента оболочки,

lcid – номер кривой изменения давления, приложенного к оболочке (по нормали),

sf – коэффициент масштаба.

6) Нагрузка на сегмент (давление)

```
*LOAD_SEGMENT
$   lcid      sf      at      n1      n2      n3      n4
    1      40.000      3001      3002      3003      3004
    1      40.000      3619      3620      3651      3650
```

7) Ускорение поступательное или угловое, прикладываемое к телу

```
*LOAD_BODY_GENERALIZED
$   n1      n2      lcid      drlcid      xc      yc      zc
    1      6      0      2      0.0      0.0      0.0
$   ax      ay      az      omx      omx      omz
    0.0      0.0      0.0      0.0      1.0      0.0
```

n1 – начальный номер узла или совокупности частей, к которым прикладывается нагрузка,

n2 – конечный номер узла, или = 0 для совокупности частей.

lcid – номер кривой,

xc, yc, zc – координаты центра вращения,

ax, ay, az, omx, omx, omz – величины ускорений.

8) Ускорение свободного падения

```
*LOAD_BODY_X, _Y, _Z
$ Gravity
$   lcid      df      lciddr      xc      yc      zc
    1 9.810E-03
```

lcid – номер кривой,

df – коэффициент масштаба.

9) Граничные условия, перемещение части

```
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$   pid      dof      vad      lcid      sf      vid      death
    2      3      2      1      -15.0
```

pid – номер части,

dof – степень свободы, = 1-2-3 – перемещения вдоль X-Y-Z, = 5-6-7 – поворот вокруг X-Y-Z (и др.),

vad, = 0 – скорость, = 1 – ускорение, = 2 – задаётся фиксированное перемещение (и др.),

lcid – номер кривой изменения координаты (при этом, повороты должны быть заданы в радианах),

sf – коэффициент масштаба,

death – время окончания действия.

10) Переменное во времени ГУ для узла

```
*BOUNDARY_PRESCRIBED_MOTION_NODE
$ node 41 is displaced in the z-direction according to load curve 1
$      nid      dof      vad      lcid      sf      vid
      41
```

11) Переменное во времени ГУ для узлов

```
*BOUNDARY_PRESCRIBED_MOTION_NODES
$      nid      dof      vad      lcid      sf      vid
      1         4         2         1 4.000E-02      1
      2         4         2         1 4.000E-02      1
      3         4         2         1 4.000E-02      1
      4         4         2         1 4.000E-02      1
```

12) Переменное во времени ГУ для узлового набора

```
*BOUNDARY_PRESCRIBED_MOTION_SET
$      nsid      dof      vad      lcid      sf      vid      death
      1         3         2         1      -15.0
      3         2         1         1         1.0
```

nid – номер узла.

13) Фиксация степеней свободы узла

```
*BOUNDARY_SPC
$      nid      cid      dofz      dofz      dofz      dofz      dofz      dofz
      1         1         1         1         0         0         1         1
```

nid – номер узла,

cid – система координат (см. *DEFINE_COORDINATE_SYSTEM в Keyword Manual),

dof = 0 – освободить степень свободы (поступательные и поворот), = 1 – закрепить.

14) Фиксация степеней свободы узлов

```
*BOUNDARY_SPC_NODE
$ 1 - fix
$      nid      cid      dofz      dofz      dofz      dofz      dofz      dofz
      3001      0         0         1         1         1         1         1
      3651      0         0         1         0         1         0         1
```

15) Начальная скорость в указанных узлах

```
*INITIAL_VELOCITY_NODE
$      nid      vx      vy      vz      vxe      vye      vze
      91 4.000E+02 0.000E+00 0.000E+00
      117 4.000E+02 0.000E+00 0.000E+00
```

16) Начальная скорость в узловом наборе

```
*INITIAL_VELOCITY
$      nsid      nsidex      boxid
      2
$      vx      vy      vz      vxr      vyr      vzr
      0.0      10.0      0.0      0.0      0.0      0.0
```

17) Начальная скорость узлов в указанной области (параллелепипеде)

```

*INITIAL_VELOCITY
$   nsid   nsidex   boxid
                        5
$   vx     vy     vz     wx     wy     wz
    0.0    0.0    -89.0
*DEFINE_BOX
$   boxid   xmm     xmx     ymn     ymx     zmn     zmx
    5       -39.0   39.0   -39.0   39.0   -25.41  51.0

```

18) Начальная скорость части или набора частей

```

*INITIAL_VELOCITY_GENERATION
$   sid     styp     omega     vx     vy     vz
$2,2,0,0,-15,0
    1       1           1.246E-01 0.000E+00-3.339E-02
$   xc     yc     zc     nx     ny     nz     phase

```

\$0,0 - ось вращения, три координаты на оси вращения и косинусы наклона оси вращения

sid – № part, styp = 2 – первый параметр определяет номер части

styp – см. Keyword Manual,

omega – угловая скорость вращения вокруг оси,

vx, vy, vz – скорости по X, Y, Z.

2.15. Связывание степеней свободы

1) Шарниры

```

*CONSTRAINED_JOINT_SPHERICAL (_REVOLUTE, _CYLINDRICAL, _PLANAR, _UNIVERSAL,
 _TRANSLATIONAL, _GEARS и т.д. (см. Keyword Manual))
$   n1     n2     n3     n4     n5     rps     damp
    99     227    0     0     0     0 0.000E+00

```

n1 – узел в недеформируемом теле A,

n2 – узел в недеформируемом теле B,

n3 – n6 – опционально, в зависимости от вида шарнира,

rps – относительная жёсткость, по умолчанию = 1.0,

damp – коэффициент демпфирования по отношению к общему.

2) Жёстко связать узловой набор 1 с частью 1

```

*CONSTRAINED_EXTRA_NODES_SET
$   pid     nsid
    1       1

```

3) Связать указанные степени свободы выбранных узлов

```

*CONSTRAINED_LINEAR
$   num
    2
$   nid     dofz     dofry     dofz     coef
    40           1           1           1.00
    42           1           1           -1.00

```

4) Связать («склеить») узел оболочки с узлом твёрдотельника

```
*CONSTRAINED_SHELL_TO_SOLID
$      nid      nsid
      326        1
```

5) Склеить совпадающие по координатам узлы с возможностью разрушения связи

```
*CONSTRAINED_TIED_NODES_FAILURE
$      nsid      eppf      etype
      1          0.0850
```

nsid – номер набора узлов,

eppf – деформация (см. Keyword Manual),

etype – тип элемента узловой группы, = 0 – оболочка, = 1 – твёрдотельный.

2.16. Контакты

1) Универсальный

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE
$1,2,3,3 - в одну строку, без трения
$      ssid      msid      sstyp      mstyp      sboxid      mboxid      spr      mpr
      1          2          3          3          0          0          0          0
$      fs        fd        dc        vc        vdc        penchk      bt        dt
      0.1500      0.000      0.000      0.127E+09      0.2000      0          0.0      0.100E+08
$      sfs       sfm       sst       mst       sfst       sfmt       fsf       vsf
      1.000       1.000       0.000       0.000       1.000       1.000       1.000       1.000
```

ssid, msid – номера контактирующих частей (slave и master), частей, сегментов, наборов узлов и т.д.,

sstyp – подчинённый тип, = 0 – сегмент (контакт поверхность-поверхность), = 1 – набор элементов оболочки (к-кт пов.-пов.), = 2 – набор частей, = 3 – часть, = 4 – набор узлов (к-кт узел-пов.), и т.д.

mstyp – мастер тип,

fs = 0.15 – статический коэффициент трения,

vc = 0.127e9 – коэффициент вязкостного трения (критический),

vdc = 0.2 – коэффициент демпфирования вязкостного трения в процентах,

dt = 10e6 – время жизни контакта по умолчанию (см. Илюшкин М.В. и др.).

2) Прочие типы контактов, включая автоматические, см. в Keyword Manual и Theory Manual

```
*CONTACT_AUTOMATIC_SINGLE_SURFACE
$ Equating ssid to zero means that all segments are included
      0
      0.08      0.08
```

```
*CONTACT_AUTOMATIC_SINGLE
      1          0          2          0
```

```
*CONTACT_AUTOMATIC_GENERAL
```

...

2.17. Управляющие параметры

1) Корректировка шага по времени в явном анализе

```
*CONTROL_TIMESTEP
$ dtinit scft isdo tslimt dtms lctm erode ms1st
, 0.05
$ .0 .9
```

scft – коэффициент безопасности, по умолчанию = 0.9 от времени, за которое звуковая волна проходит элемент (подробнее см. CFL), для повышения устойчивости при плохом качестве КЭ-сетки можно назначить = 0.01.

2) Время моделирования процесса

```
*CONTROL_TERMINATION
$ endtim endcyc dtmin endneg endmas
3.000E-02
```

3) Активация адаптивного перестроения сетки (части модели, сетка которых перестраивается адаптивно, задаются картой *PART)

```
*CONTROL_ADAPTIVE
$ adpfreq adptol adpopt maxlvl tbirth tdeath lcadp ioflag
5.0e+0 0.1 2 2 0.0 0.0 0
$0.01,10,2,2
```

adpfreq – промежуток времени между адаптивными перестроениями сетки,
 tbirth, tdeath – моменты времени, в которые начинается и заканчивается адаптивное перестроение сетки,
 остальные параметры см. в Keyword Manual.

4) Объёмная вязкость

```
*CONTROL_BULK_VISCOSITY
$ Q1 Q2 type
1.500 .060
```

Q1 – квадратичный коэффициент вязкости, по умолчанию = 1.6,
 Q2 – линейный коэффициент вязкости, по умолчанию = 0.06.

5) Конструкционное демпфирование

```
*CONTROL_DAMPING
$ nrcyck drtol drfctr drterm tssfdr irelal edttl idrflg
100 1.0e-3 0.995 0.9
*DAMPING_GLOBAL
$ lcid valdmp
$ 0 10.0
3
```

6) Диссипация энергии

```
*CONTROL_ENERGY
$ 2 - учёт деформации элементов по типу песочных часов
$ hgen rwen slnten rylen
2 1 2 2
```

hgen = 1 – энергия не рассчитывается (по умолчанию), 2 – энергия рассчитывается и включается в полный баланс энергии,
 rwen – см. Keyword Manual,
 slnten – диссипация контактной энергии, =1 – отключена, = 2 – рассчитывается,
 rylen – рэлеевское рассеивание энергии (с затуханием), = 1 – отключено, = 2 – включено.

7) Неустойчивость формы конечного элемента по типу «песочных часов»

```
*CONTROL_HOURGLASS
$      ihq      qh
      1      .100
*HOURGLASS
$      i      i      f      i      f      f
$      hgid      ihq      qm      ibq      q1      q2
      1              0.05      0.06      1.2
```

hgid – порядковый номер подавления,
 ihq – тип подавления, = 1 по умолч,
 qh, qm – коэффициент подавления (>0.15 может стать причиной неустойчивого решения),
ВНИМАНИЕ: см. Keyword Manual.

2.18. База данных результатов

1) Шаг по времени для записи результатов на диск

```
*DATABASE_BINARY_D3PLOT
$ dt/cycl      lcdt - time interval between state dumps (D3PLOT)
  5.000E-04
```

2) Шаг записи результатов для предварительно выбранных элементов, узлов, частей

```
*DATABASE_BINARY_D3THDT
$      dt      lcdt
$ DT/CYCL
  999999
$ .000E+00
```

3) История выбранных узлов, оболочек, твёрдотельников, шарниров

```
*DATABASE_HISTORY_NODE
$ Define nodes that output into nodout
$      id1      id2      id3      id4      id5      id6      id7      id8
      22      23      35      36
*DATABASE_HISTORY_SHELL
$      id1      id2      id3      id4      id5      id6      id7      id8
      1
*DATABASE_HISTORY_SOLID
$      id1      id2      id3      id4      id5      id6      id7      id8
      1
*DATABASE_JNTFORC
$      dt
      0.0001
```

4) Группа узловых сил

```
*DATABASE_NODAL_FORCE_GROUP
$      nsid      cid
      201
```



```
*DATABASE_NODFOR
$ dt
0.010
```

5) Формат сохраняемой базы данных

```
*DATABASE_FORMAT
$ IFORM
0
```

IFORM = 0 – LS-DYNA (по умолчанию), = 1 – ANSYS, = 2 – ANSYS и LS-DYNA.

6) Шаг выдачи глобальной статистики в лог-файл

```
*DATABASE_GLSTAT
$ dt
2.000E-04
```

7) Шаг выдачи суммарной энергии системы

```
*DATABASE_MATSUM
$ dt
2.000E-04
```

2.19. Дополнительные ключевые слова

1) Переключить анализ на неявный

```
*PARAMETER
$ начальный шаг по времени, минимальный шаг, максимальный шаг
rdt 1e-4
rdtmin 1e-4
rdtmax 2e-4
*CONTROL_IMPLICIT_AUTO
$# iauto iteopt itewin dtmin dtmax dtexp kfail kcycle
1 20 4&dtmin &dtmax 0.0 0 0
*CONTROL_IMPLICIT_DYNAMICS
$# imass gamma beta tdybir tdydth tdybur irate
1 0.5 0.25 0.01.00000E281.00000E28 0
*CONTROL_IMPLICIT_GENERAL
$# imflag dt0 imform nsbs igs cnstn form zero_v
1&dt 1 1 2 0 0 0
*CONTROL_IMPLICIT_SOLUTION
$# nsolvr ilimit maxref dctol ectol rctol lstol abstol
4 4 7 0.001 0.01 0.0 0.01.0000E-20
$# dnorm diverg istif nlprint nlnorm d3itctl cpchk
1 1 0 2 2 0 0
$# arcctl arcdir arclen arcmt h arcdmp arcpsi arcalf arctim
0 0 0.0 1 2 0 0 0
$# lsmt d lsdire irad srad awgt sred
5 2 0.0 0.0 0.0 0.0
```

2) Указать, что выбранные части станут недеформируемыми

```
*DEFORMABLE_TO_RIGID
$ pid mrb
1
```

3) Включить ключевые слова из текстового файла с именем FILE1 (на последней строке текстового файла стоит ключевое слово *END)

```
*INCLUDE
Z:\WORK\FILE1
```

4) Время останова, заменяет значение, указанное в карте *CONTROL_TERMINATION

```
*TERMINATION
&newtime
```

5) Вектор (несколько векторов) по координатам двух точек

```
*DEFINE_VECTOR
$      vid      xt      yt      zt      xh      yh      zh
      1 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00-1.000E+00
*DEFINE_COORDINATE_VECTOR
$      cid      xx      yx      zx      xv      yv      zv
      1          1.      0.      0.      0.      1.      0.
      2 0.99144 -0.13053 0.00000 0.13053 0.99144 0.00000
```

6) Сохранять в базе данных результатов переменные истории для твёрдотельников и оболочек

```
*DATABASE_EXTENT_BINARY
$ N - количество переменных (т.е. сохранится 1-я, 2-я, ..., N-я переменная)
$#  neiph      neips      maxint      strflg
      &N          &N          &N          1
```

2.20. Принудительный останов решателя, рестарт решателя

Некорректный путь останова задачи — это завершать процесс в планировщике ОС, например убивать процесс солвера через Диспетчер задач в семействе ОС Windows, или воспользоваться командой kill в ОС, построенных на ядре Linux. Корректно остановить солвер можно следующими способами: создать файл D3KIL в рабочей папке задачи; в консоли работающего солвера нажать клавиатурную комбинацию Ctrl-C, солвер встанет на паузу, после чего набрать sw1 или quit, подробнее см. пункт Sense switch controls в Keyword Manual.

При наличии в расчётном файле ключевого слова *DATABASE_BINARY_D3DUMP LS-DYNA в процессе счёта автоматически сбрасывает на диск дампы оперативной памяти, что даёт возможность продолжить решение с произвольного шага, а также стартовать с произвольного шага после внесения на данном шаге правок в КЭ-модель и задания новых настроек. Если указанное ключевое слово на задано, после завершения всех этапов расчёта автоматически генерируется единственный дампы-файл с именем D3DUMP01 [2].

Основные причины использования рестарта: расчёт разбит на технологические этапы; требуется продолжение расчёта, прерванного в результате ошибки ОС или пользователя; продолжение расчёта после автоматического останова по достижению заданного времени счёта; при постпроцессинге было обнаружено, что анализ некорректен, требуется коррекция модели; предыдущее решение не запускалось на счёт в течение продолжительного времени; модель частично разрушилась, требуется исключить часть элементов и материалов [32]. См. также описание параметра Q=remar при необходимости сгущения сетки из твёрдотельников.

Рестарт принято делить на три типа. При **простом рестарте**, когда решатель был принудительно остановлен до момента исчерпания времени останова, LS-DYNA запускается с параметром R=rtf и продолжает вычисления из точки приостановки до заданного времени останова. В дополнительных ключевых словах нет необходимости.

Малый рестарт предусматривает продолжения расчёта с шага nn, дампы памяти которых ранее были сброшены в файлы D3DUMPnn. Опционально имеется возможность изменить настройки решателя и проекта с помощью подключения файла restartinput.k, в котором содержатся специальные ключевые слова (см. Keyword Manual). Запуск LS-DYNA производится с параметрами «I=restartinput.k R=D3DUMPnn», где R=D3DUMPnn указывает номер дампа.

Деформации и напряжения с предыдущего шага не учитываются, нельзя изменить геометрию модели. Все внесённые в задачу изменения отразятся на содержимом последующих дампов-файлов рестарта, т.е. дампов-файлы перезапишутся, в то время как база данных результатов продолжит пополняться.

Малый рестарт позволяет переустановить время останова, переустановить частоту вывода на печать, переустановить частоту вывода графиков, уничтожить контактные поверхности, уничтожить элементы и части расчётной модели, назначить деформируемое тело недеформируемым и наоборот, изменить опции демпфирования и т.д.

Для применения малого рестарта в файл restartinput могут быть записаны следующие ключевые слова: *CHANGE_, *CONTROL_CPU, *CONTROL_DYNAMIC_RELAXATION, *CONTROL_TERMINATION, *CONTROL_TIMESTEP, *DAMPING_GLOBAL, *DATABASE_, *DATABASE_BINARY_, *DEFINE_OPTION, *DELETE_, *INTERFACE_SPRINGBACK, *KEYWORD, *RIGID_DEFORMABLE_, *SET_, *STRESS_INITIALIZATION_, *TERMINATION_, *TITLE.

Полный рестарт предназначен для внесения многочисленных изменений и предоставляет возможность начать анализ со взятыми с предыдущего шага деформациями и напряжениями, задавать новые контактные пары, менять имя базы данных результатов, проводить многостадийную формовку листовых металлов и т.п. Например: выполнен расчёт файла sample1.k, был сгенерирован дамповый файл d3dump01. Создаём новый расчётный файл sample2.k, который содержит полную модель в её недеформированном исходном состоянии (исходные координаты узлов), при этом в геометрию модели нельзя вносить кардинальные изменения, т.е. нельзя удалять/добавлять узлы, классические и дискретные элементы, части, но разрешено менять начальные и граничные условия и изменять номера частей. Запускаем рестарт с параметрами I=sample2.k R=d3dump01, время счёта продолжится с шага nn (обнулять время не требуется). В соответствии с данными дампа и заданными ключевыми словами решатель задаст новые исходные положения и скорости узлов сетки, напряжения и деформации в элементах, изменит материал детали или его константы и т.п. Станут неподвижными части, материалы которых не были инициализированы, в них не будут заданы деформации и напряжения.

Примечание: если инициализированные и неинициализированные материалы имеют общие узлы, то узлы инициализированного материала будут двигаться, вызывая неожиданные деформации в неинициализированном материале.

Основное отличие полного рестарта от малого рестарта – применение ключевого слова *STRESS_INITIALIZATION. Начальное НДС можно внести во всю модель и в отдельные части модели. Подобным же образом задаётся поле скоростей ключевым словом *CHANGE_VELOCITY_, а также прочие начальные и граничные условия, см. ключевые слова *CHANGE_. В полном рестарте доступны ключевые слова: *BOUNDARY_CONDITION, *CONTACT_SMALL_PENETRATION, *CURVE_DEFINITION, *RIGID_BODY_CONSTRAINT, *RIGID_BODY_STOPPER, *STATUS_REPORT_FREQUENCY, *THERMAL_PARAMETERS, *VELOCITY, *VELOCITY_NODE, *VELOCITY_RIGID_BODY, *VELOCITY_ZERO.

Часть вторая

Хуже всего на свете ждать и догонять. Их ждут, они догоняют.

Аркадий и Борис Стругацкие. Страна багровых туч.

3. Диаграмма предельных деформаций (FLD)

3.1. Моделирование листовой формовки

Главная цель моделирования процесса формовки методами вычислительной механики – подобрать технологический режим формовки листа с минимизацией отбраковки и максимизацией вытяжки при допустимом уровне утонения, что требуется для уменьшения издержек, увеличения коэффициента загрузки оборудования и повышения рентабельности предприятия (см. пример подбора технологического режима, геометрии матрицы, пуансона и листа [55]).

Приведём последовательность этапов моделирования листовой формовки из практики BMW [51]. На рис. 1 изображена последовательность действий для получения из листа элемента силового каркаса авто.

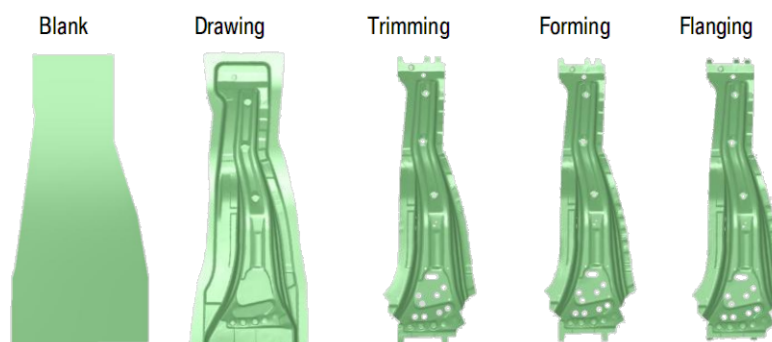


Рис. 1. Вытяжка, обрезка, формовка и догибка листового изделия. Источник: [51]

На рис. 2 показана одна из итераций цикла уточнения технологического режима формовки. В процессе моделирования оцениваются НДС, утонение, поводки конструкции от пружинения, жёсткость конструкции и т.д. Здесь под пружинением (springback) понимается процесс самостоятельного изменения формы изделием из-за остаточных несбалансированных напряжений [54].

На рис. 3 поэтапно расписан процесс подготовки к компьютерному моделированию. Он предусматривает:

- определение механических констант реальных образцов в условиях лаборатории;
- подбор по экспериментальным кривым подходящего закона поведения материала с учётом особенностей пластического деформирования;
- верификацию подобранного закона поведения и механических констант с помощью численных экспериментов, являющихся аналогами проведённых ранее механических испытаний;

- внесение значений констант материалов в шаблоны ключевых слов для включения в расчётные файлы (создание библиотеки материалов);
- моделирование процессов листовой формовки для предварительной оценки ожидаемого уровня качества проектируемого изделия в условиях заданной производственной технологии.

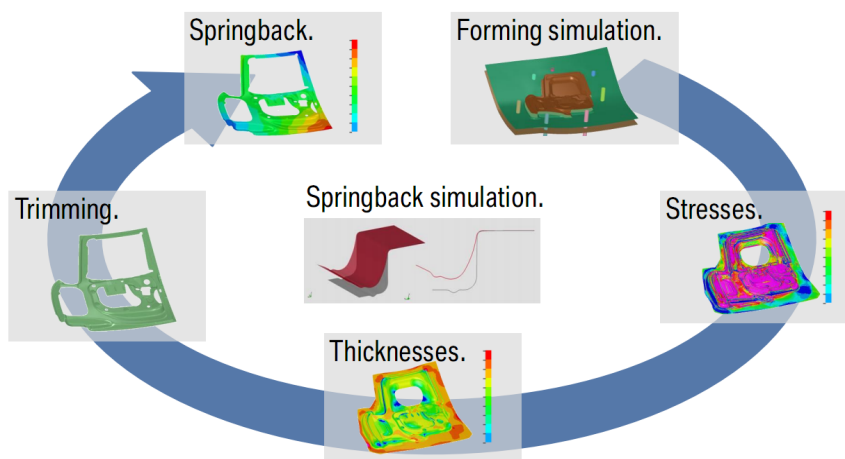
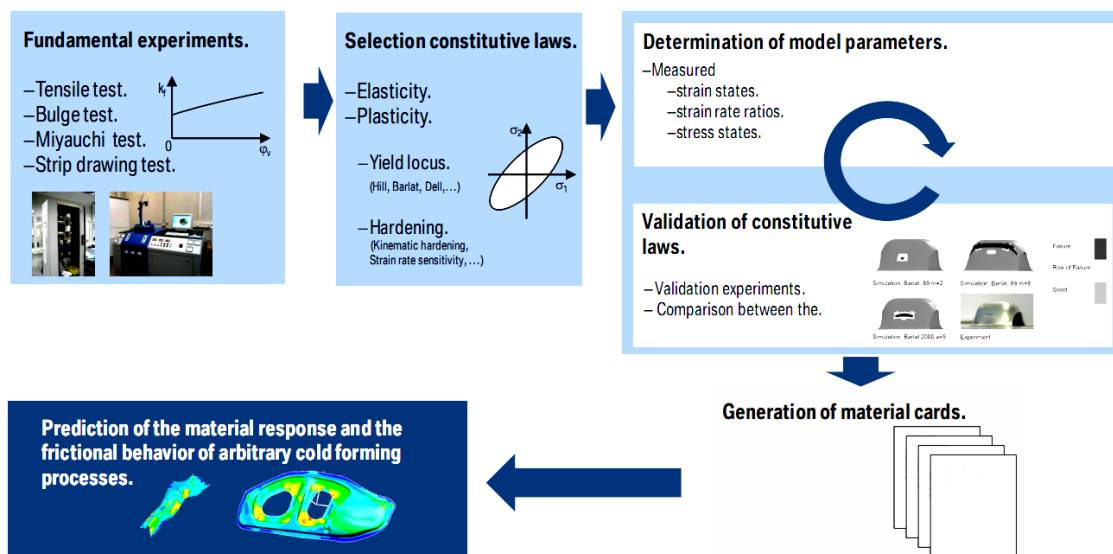


Рис. 2. Цикл моделирования формовки. Источник: [51]



Usage of LS-DYNA in metal forming. 10th European LS-DYNA Conference – 15. - 17.6.2015.

Page 12

Рис. 3. Этапы подготовки к компьютерному моделированию. Источник: [51]

Оператор программного комплекса выбирает модель поведения материала, вносит в вычислительную модель геометрию инструмента (матрица, пуансон) и листа, константы из базы данных материалов, подбирает характеристики трения. Далее солвер на основе входных данных о материалах, геометрии, граничных условий вычисляет напряжения, деформации и перемещения [34].

Единственное, остаётся внести в модель критерий разрушения⁶. На производстве в качестве данного критерия может применяться диаграмма FLD, где указаны предельные уровни деформаций. При моделировании главные деформации в модели формируемой детали сравнивают с FLD: если найденные деформации падают в безопасную зону, то на производстве формовка реального металла пройдёт успешно. FLD применима только для тех операций, в которых история нагружения не играет значимой роли, процесс нагружения является монотонным, а напряжённое состояние — плоским, а именно — формовка, вытяжка, гибка, обжим, раздача [60].

6 Планируется выпуск методички за авторством: Келлер И.Э., Трофимов В.Н., Петухов Д.С., Казанцев А.В.

3.2. Диаграмма предельных деформаций Forming-limit diagram (FLD)

Диаграмма FLD строится⁷ в осях $\epsilon_{min} - \epsilon_{max}$ и показывает уровень критических главных деформаций листа из металла конкретной партии, определяется экспериментально. Площадь под кривой FLD охватывает все возможные комбинации деформаций, которые лист выдерживает без разрушения, но с возможным складкообразованием. Из практики известно, что утонение может случаться под FLD, поэтому на основе опытных производственных данных на диаграмме отмечаются границы безопасной и предельной зон [34 С. 1728, С. 1973, С. 2002]. Примеры аппроксимированных по точкам кривых FLD показаны на рис. 4.

FLD получают теоретически, экспериментально, либо численными методами. При отсутствии опытных данных вычисляется **теоретическая FLD** для данного листа (например, см. [39-41,44]), причём теоретические методы далеко не всегда дают достоверную кривую FLD (см. [41,45,47]).

Наиболее точен чисто натурный эксперимент, несмотря на свою дороговизну и временные затраты. Для построения FLD листы металла деформируют различными способами (подробный обзор [42]). В основном применяют тест на вытяжку вдавливанием стальной полусферы (runch test), который учитывает влияние трение между листом и оснасткой, или гидравлическую формовку.

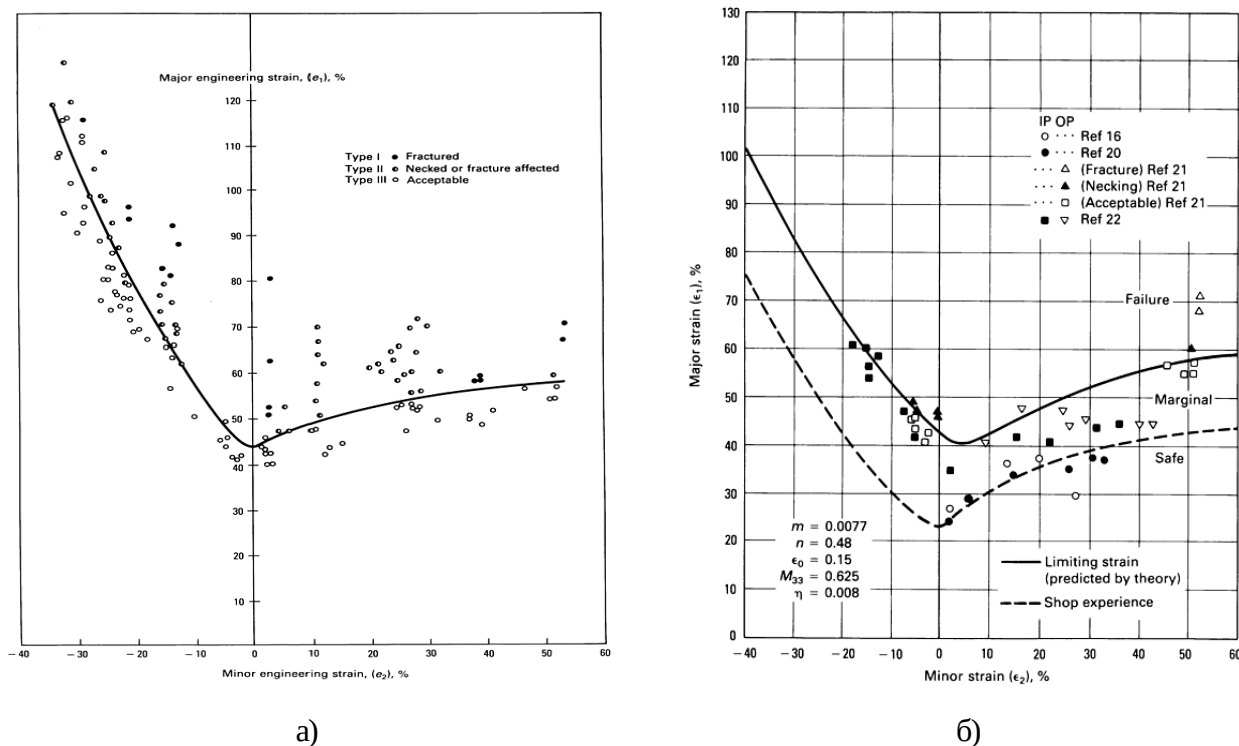


Рис. 4. Примеры FLD для спокойных сталей (aluminum-killed steel): а) эксперимент; б) сплошная линия найдена теоретически; прерывистой линией обозначена граница зоны безопасности, найденная на практике; между прерывистой и непрерывной линией находится пограничная зона. Источник: [34]

7 О том, как строить FLD в LS-PREPOST, см. <http://www.lstc.com/lsp/content/pages/1/fld/fld.shtml>

В любом экспериментальном методе перед формовкой на лист металла наносится сетка концентрических окружностей⁸, принимается гипотеза о ПНС; например, стандарт ASTM E 2218-02 предусматривает ячейки сетки диаметрами 2.5 мм с нанесением сетки методами фотопринта, фототравления или электроэрозией; в работе [38] предлагается наносить кружки диаметрами 8-10 мм, при этом исчезает ряд недостатков, присущих стандарту.

Лист металла рубится для создания концентратора определённой формы (см. ниже). Далее образец фиксируется прижимом к матрице с круглым отверстием, металлический пуансон штампует пластинку до её разрушения. В узких пластинках разрушение наступает при отношении $\varepsilon_{minor}/\varepsilon_{major} \approx -0.5$, что сравнимо со значением, получаемым в эксперименте на растяжение. Наглядные фотографии разрушенных пластинок см., например, в [37].

После необратимой пластической деформации окружности на поверхности образца переходят в эллипсы, за исключением зон с двухосным растяжением. Деформации измеряются в зонах видимого разрушения, утонения, шейкообразования [46] и вокруг них. Большие и малые оси эллипса сравниваются с диаметром первоначальных окружностей, в каждом эллипсе находят главные деформации (major and minor strains), их комбинации вносят на диаграмму FLD. Контур FLD рисуется над измеренными деформациями вне областей с утонением и под областями с разрушением и утонением, см. рис. 4, б).

Коротко о численном методе построения FLD [38]. На основе нескольких натуральных испытаний, например, на одноосное растяжение, статистически подбирается закон $\sigma - \varepsilon$ (степенной и т. п.), его найденные константы подставляются в расчётный файл LS-DYNA, где формируется модель punch-теста; допускается гипотеза о ПНС. За критический уровень деформаций в вычислительном эксперименте принимают деформации начала разрушения элементов или деформации, предшествующие в процессе необратимого утонения элементов. Деформации в шейке образца, рядом с трещиной или зоной утонения вносят на FLD.

Численные методы (см. пример [43]) опираются на качество экспериментальных данных, а именно на значения тех материальных констант, которые закладываются в численную модель; отсюда чем больше элементарных экспериментов будет проведено с натурными образцами, тем сложнее и точнее может быть модель поведения материала в вычислительном эксперименте. К примеру, ряд натуральных экспериментов позволяет оценить анизотропию свойств материала, без учёта которой численные методы могут сдвигать кривую FLD [37,38], что скажется на качестве прогнозирования состояния изделия.

3.3. Области FLD

Область положительных главных минимальных деформаций соответствует двухосному растяжению, область отрицательных – одноосному растяжению. Случаи, при которых минимальные главные деформации равны нулю, соответствуют НДС. Форма и расположение FLD уникальна для каждого материала. FLD действительна в тех случаях, когда напряжённое состояние является плоским, а отношение главных пластических деформаций постоянно в течение всего процесса деформации, то есть деформация является монотонной, как в операциях листовой штамповки (а именно при формовке, вытяжке, гибке, обжиге и раздаче) [60].

Варьирование ширины галтелей и радиусов вырезов в образцах требуется для построения FLD по точкам, т.к. позволяет получить различные НДС в заготовке при деформировании (см. рис. 5, 6). Образец без боковых вырезов позволяет получить точку на кривой FLD, которая соответствует двустороннему растяжению $\varepsilon_1 = \varepsilon_2$; образец с максимальными вырезами позволяет получить точку, которая приблизительно соответствует одноосному растяжению $\varepsilon_1 = -2\varepsilon_2$ [38].

8 См. Сурсыков В.А. Определение деформаций по искажению делительных сеток: Методическая разработка. - Пермский государственный технический университет. - Пермь, 2003. - 15 с.

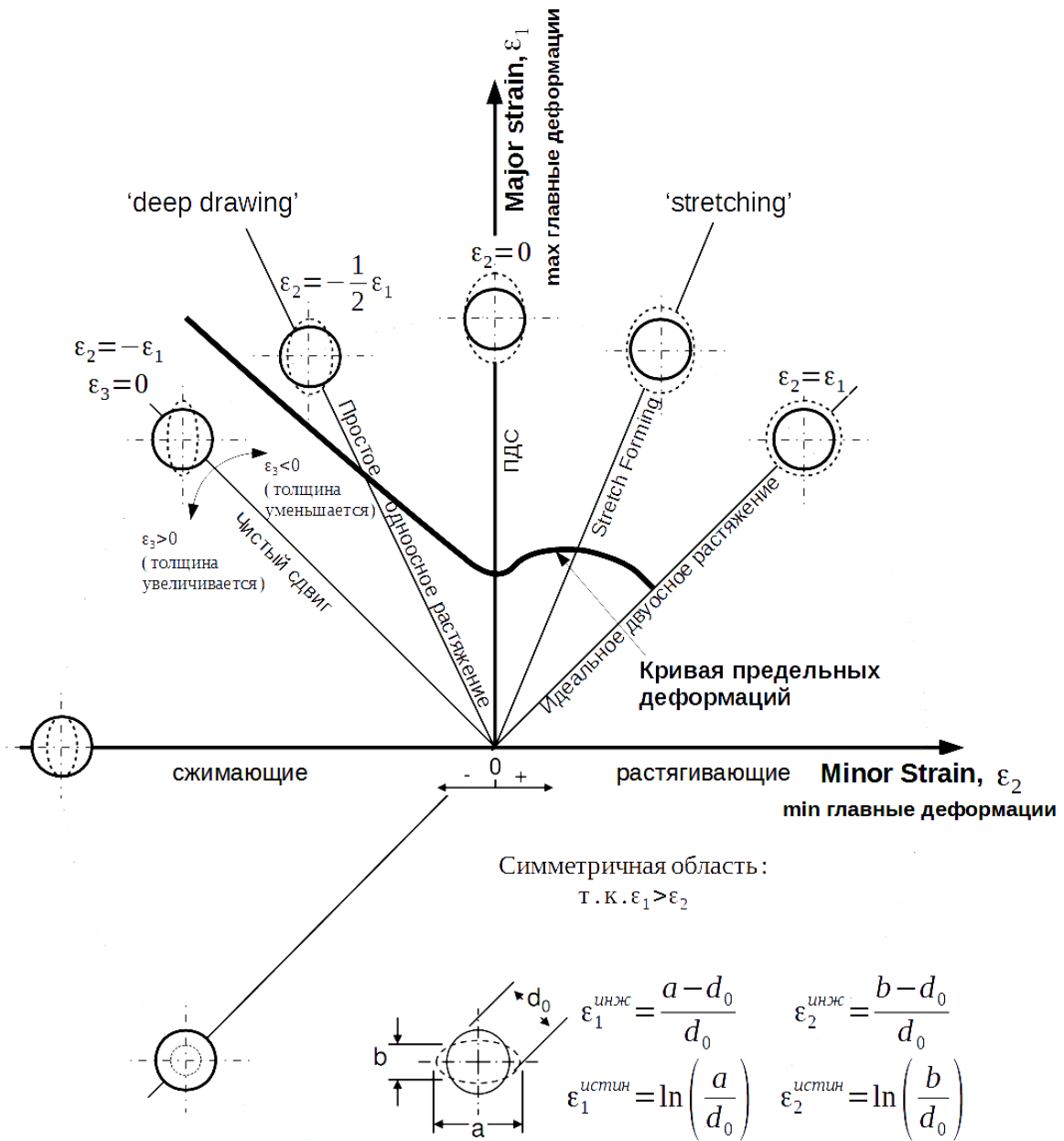


Рис. 5. Области классической предельной диаграммы. Источники: [41, 59, 62, 68, 70]

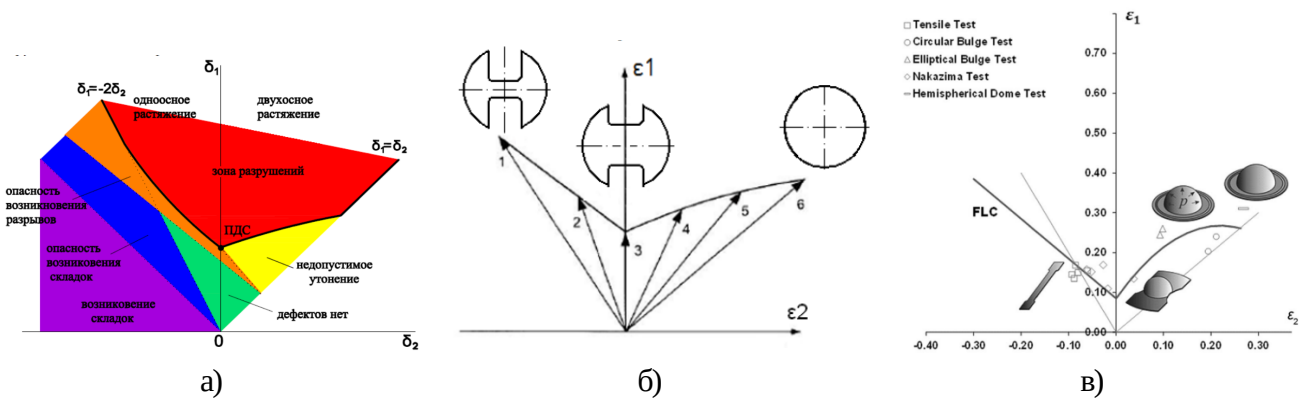



Рис. 6. FLD: а) области диаграммы; б) варианты заготовок в тесте Наказимы для получения различных деформированных состояний; в) FLD для листов из сплава алюминия AA1050 толщиной 1 мм. Источники: [43, 46, 60]

3.4. Оценка предельной деформации в LS-DYNA

В постпроцессоре LS-DYNA диалог работы с диаграммой предельных деформаций вызывается после нажатия на характерную иконку . Поясим основные пункты окна FLD (см. рис. 7).

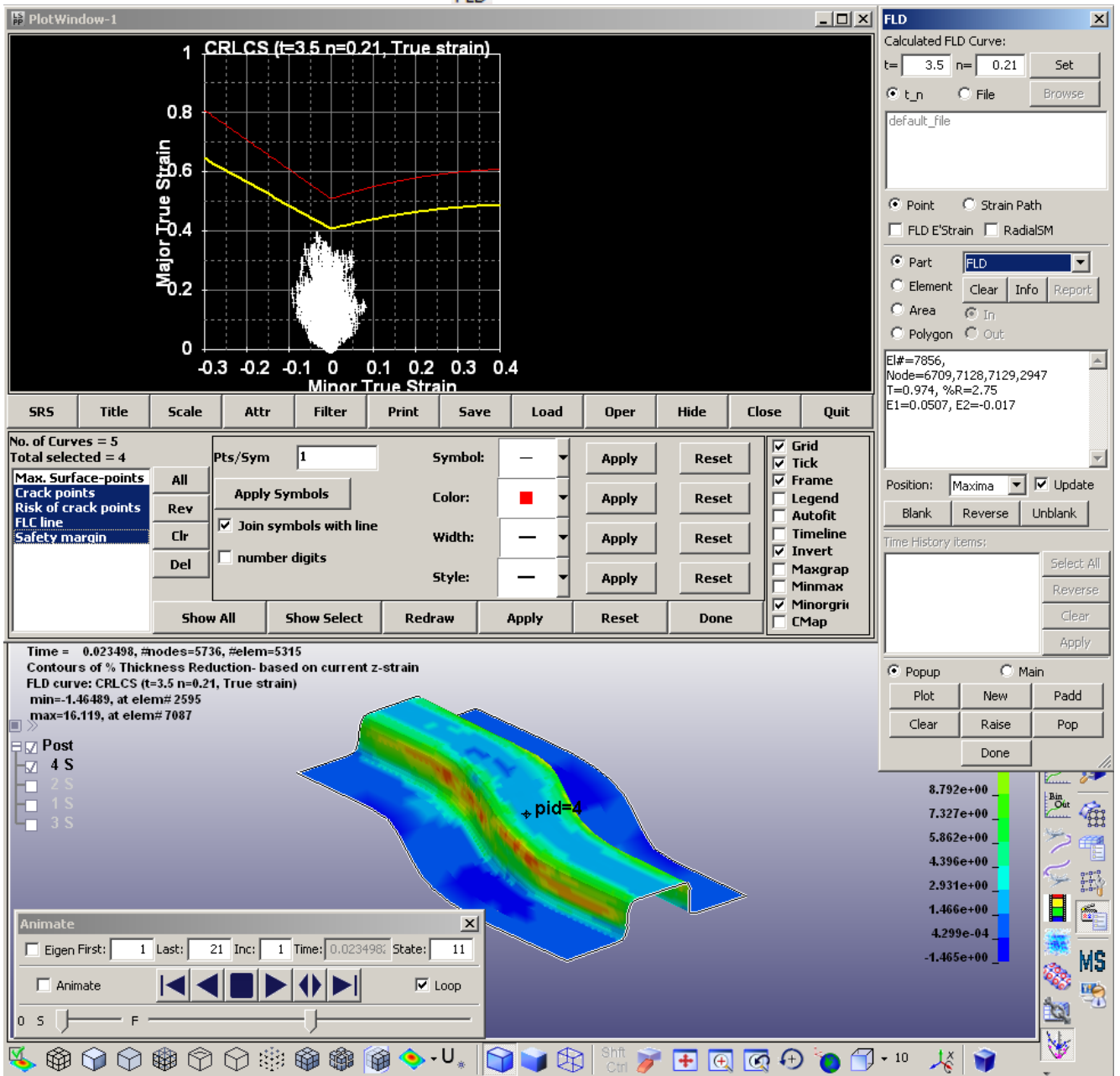


Рис. 7. Постпроцессинг FLD. Источник файла-примера: [56]

Строковые поля “t=” и “n=” отвечают за вид теоретических кривых, одна из которых отображается на диаграмме по умолчанию красным цветом и отвечает за деформации разрушения, другая – жёлтым, и показывает границу зоны теоретических безопасных деформаций. При этом, t – толщина листа, n – показатель экспоненты (strain hardening exponent) в законе упрочнения вида:

$$\sigma = K \varepsilon^n \tag{15}$$

Показатель n можно считать константой материала, если не требуется учитывать температуру. Приведём значение n для некоторых сплавов по классификации, принятой в США⁹:

9 https://en.wikipedia.org/wiki/Strain_hardening_exponent

- закалённая сталь с низким содержанием углерода = 0.26;
- сталь 4340 отпуск при 315 °С = 0.15;
- закалённая нержавеющая сталь 304 = 0.45;
- закалённая медь = 0.54;
- закалённая корабельная латунь = 0.49;
- термообработанный сплав алюминия 2024-T3 = 0.16;
- закалённый сплав магния AZ-31B = 0.16.

Радио-кнопки “t_n” и “File” выбирают теоретическую кривую или реальную, взятую из файла. Пример содержимого текстового файла с данными реальной кривой FLD [56]:

```
FLD curve (true strain)
11
-0.1324, 0.2624
-0.0834, 0.2461
0.0000, 0.2159
0.0218, 0.2078
0.0411, 0.2038
0.0488, 0.2054
0.0788, 0.2247
0.1017, 0.2390
0.1284, 0.2476
0.1807, 0.2624
0.2327, 0.2647
```

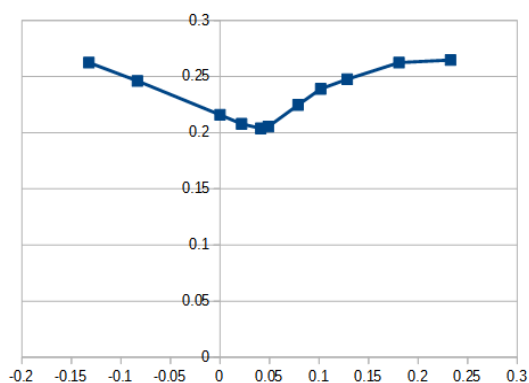


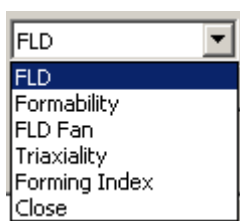
Рис. 8. Графическое представление данной кривой

Далее, радио-кнопка “Point” - разместить на диаграмме точки.

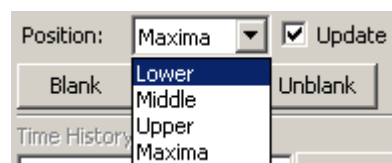
Флаг “FLD E’Strain” при активации переключает шкалы деформаций FLD с логарифмических деформаций к стандартным инженерным.

Радио-кнопки “Part”, “Element”, “Area” и “Polygon” отвечают за выбор объекта, деформации в конечных элементах которых попадут на диаграмму. Выбор объектов осуществляется кнопками мыши.

Далее выбирают форму диаграммы. Для отображения и теоретической FLD, и текущего уровнем деформаций в выбранных элементах выберите в списке строку FLD (см. рис. 9 а)) и нажмите кнопку Plot (см. рис. 7).



а)



б)

Рис. 9. а) выбор диаграммы; б) выбор точки интегрирования.

При выборе в выпадающем списке пункта Formability строится расширенная теоретическая диаграмма (см. рис. 10), при этом формируемая деталь разбивается на зоны в зависимости от их состояния: максимальная вероятность появления трещин, риск трещины, склонность к утонению, норма, чрезмерная вытяжка, тенденция к складкообразованию, складки.

Внимание: Подстановка реальных значений критериев трещиностойкости, утонения и складкообразования целиком и полностью осуществляется оператором LS-DYNA; значения эти необходимо подбирать по данным экспериментальных исследований с натурными образцами.

Пункт Triaxiality используется концерном Daimler, подробнее см. [50, 57, 65, 66, 67].

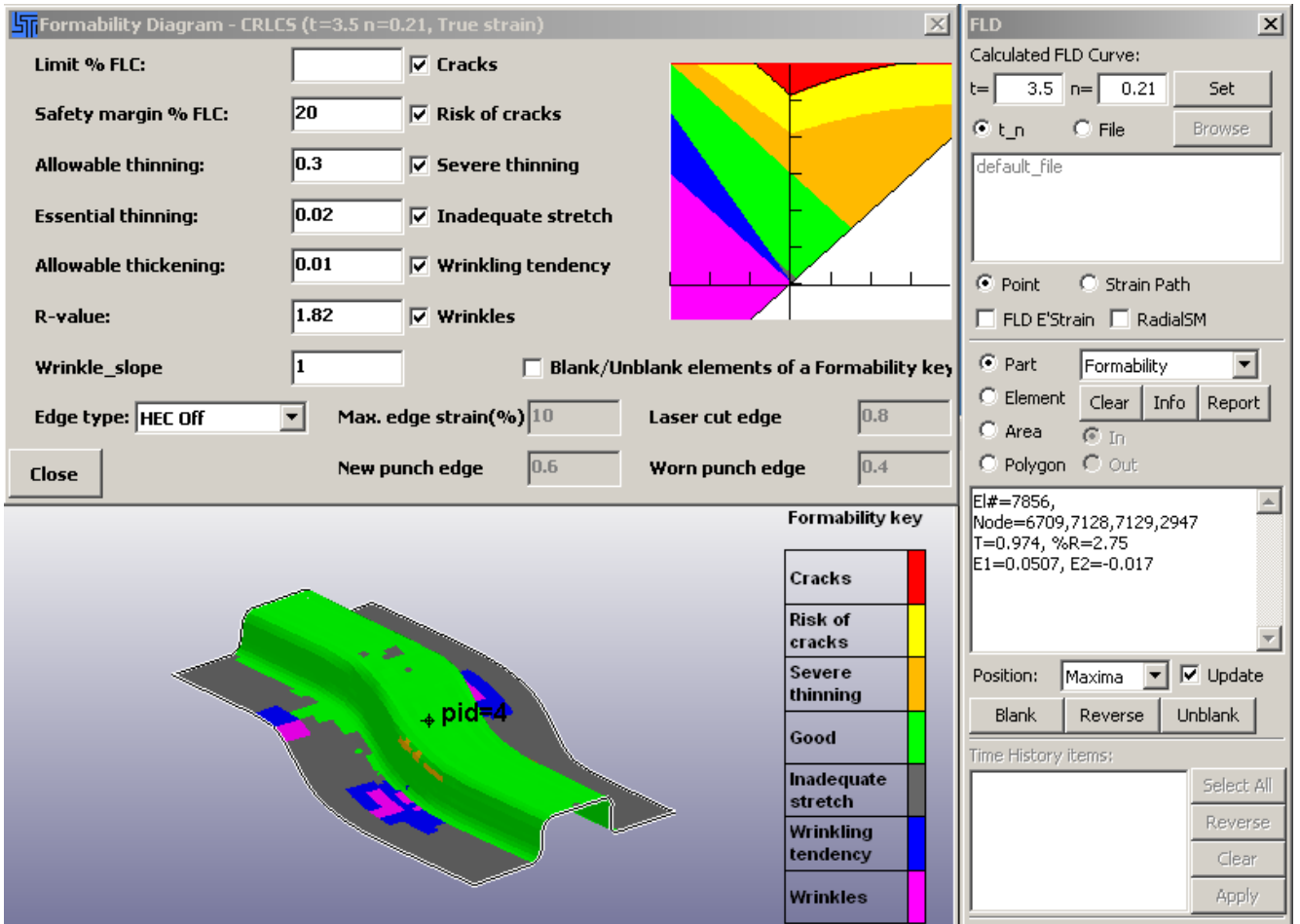


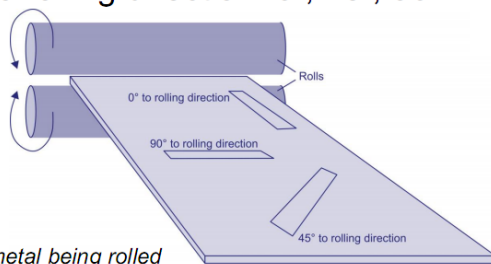
Рис. 10. Расширенная теоретическая диаграмма и текущее состояние конечных элементов формируемой детали

Раскрывающийся список “Position” отвечает за выбор точки интегрирования в конечных элементах: на поверхностях элементов, в срединном слое, или выбирается максимальные по модулю деформации. Флаг “Update” позволяет увидеть динамику FLD во время анимации.

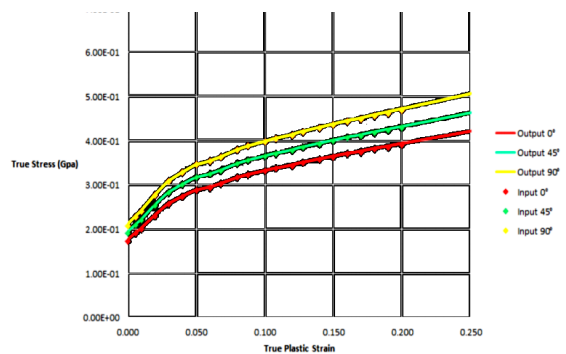
В окне Formability Diagram – CRLCS коэффициент R-value – это коэффициент Ланкфорда, определяющий анизотропное пластическое поведение материала¹⁰.

3.5. Анизотропия листового металла

- Angles to rolling direction: 0°, 45°, 90°



а)



б)

Рис. 11. Влияние анизотропии листа на кривые растяжения: а) образцы, вырезаемые из катанного листа под углами 0, 45 и 90 градусов, источники [30, 63]; б) кривые растяжения для образцов, вырезанных из листа под углами 0, 45 и 90 градусов к направлению прокатки, источники: [49, 52, 64]

10 См. https://en.wikipedia.org/wiki/Lankford_coefficient

3.6. Задание: построить FLD по данным вычислительного эксперимента

Провести серию экспериментов по определению FLD на основании ISO 12004-2:2008 “Determination of Forming Limit Curves in the Laboratory” [43, 58, 69].

Геометрия оснастки для теста Наказимы:

- диаметр полусферического индентора 100 мм;
- внутренний диаметр матрицы 105-108 мм;
- радиус фаски матрицы 5-10 мм;
- коэффициент кулоновского трения 0.04-0.08;

Геометрия оснастки для теста Марсиньяка и модифицированного теста Наказимы:

- внешний диаметр цилиндрического (Марсиньяк) и полусферического (Наказима) индентора 100 мм ;
- внутренний диаметр цилиндрического индентора 33 мм (Марсиньяк);
- внутренний диаметр матрицы 120 мм;
- радиус фаски матрицы 10-20 мм;
- радиус фаски индентора 10 мм (Марсиньяк);
- коэффициент кулоновского трения 0.15.

Принять следующие параметры процесса:

- толщина листа заготовки 1.2, 2.5 мм;
- диаметр листа заготовки 200 мм;
- усилие пуансона 100 кН (600кН для модифицированного теста Наказимы);
- скорость движения пуансона 1.5 ± 0.5 мм/с (4 мм/с);
- тип элемента - оболочка, формулировка № 16 (полное интегрирование), количество точек интегрирования 5 или 9;
- линейный размер конечного элемента 2-3 мм;
- материал листовой заготовки сплав AA2024-O [58], использовать *MAT_3-PARAMETER_BARLAT (см. Keyword Manual и Material Manual [2,4]):

Предел текучести (YS), МПа	(UTS), МПа	Модуль Юнга, МПа	Коэффициент Пуассона	R0	R45	R90	K, МПа	n	Плотность, ρ , кг/мм ³
72	121	73100	0.33	0.65	0.83	0.6	326.8	0.226	2.73e-9

- материал листовой заготовки сплав AA5182-O [72], использовать *MAT_3-PARAMETER_BARLAT, кривая упрочнения на рис. 12:

Модуль Юнга, МПа	Коэффициент Пуассона	R0	R45	R90	m	Плотность, ρ , кг/мм ³
69000	0.33	0.85	0.77	0.9	8	2.7e-9

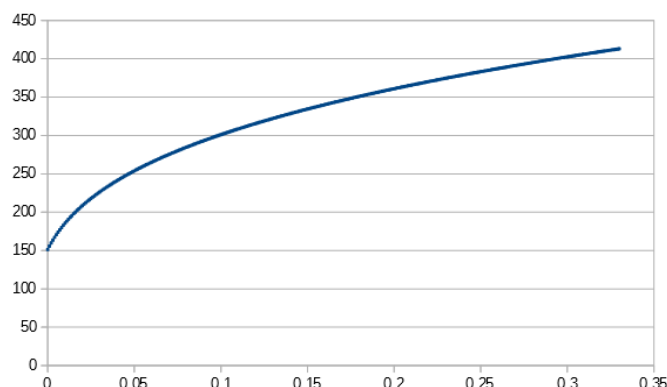


Рис. 12. Зависимость $\sigma(\epsilon)$ для AA5182-O. Источник: [72]

- варианты геометрии заготовок представлены на рис. 13, для метода Наказимы взять в расчёт и цельную заготовку.

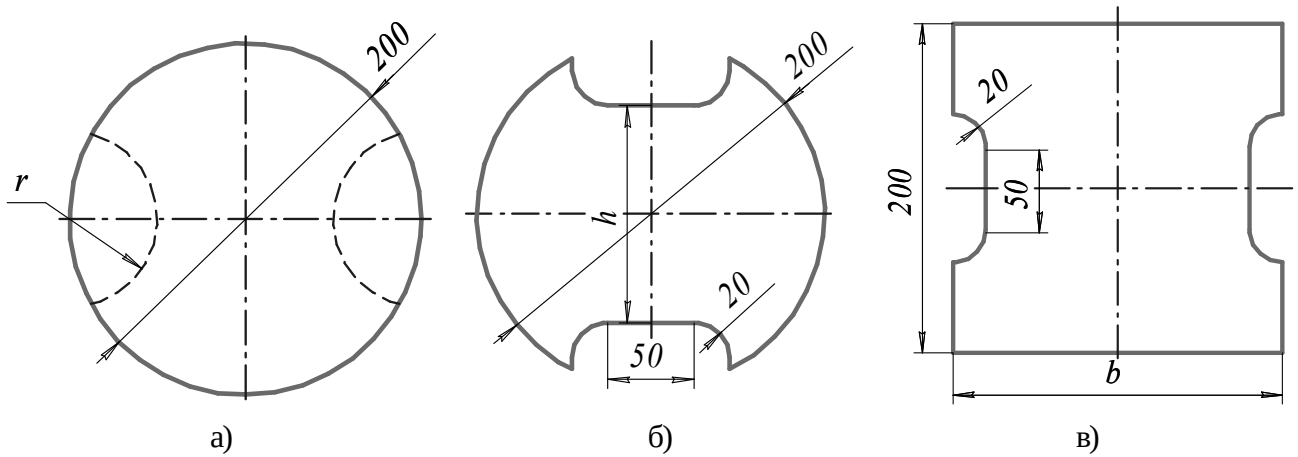


Рис. 13. Геометрия заготовок: а) метод Хейсика; б) метод Марсиньяка; в) модифицированный метод Наказимы. Размеры r , h , b см. в табл. 4. Источник: [43]

Таблица 4. Размеры заготовок

Метод	Параметр	Значения параметра										
		0 (без вырезов)	85	75	65	60	50	40	30	-	-	-
Хейсик [58]	r , мм	0 (без вырезов)	85	75	65	60	50	40	30	-	-	-
Марсиньяк [43]	h , мм	200 (без вырезов)	160	155	145	135	125	115	95	75	55	35
Наказима [41, 43, 61, 68]	b , мм	200	190	180	170	145	130	115	100	85	70	55

Добавить ключевые слова для сохранения НДС и толщин оболочек в файле результатов, а также для учёта влияния изменения толщин на НДС:

```
*DATABASE_EXTENT_BINARY
, , 3, 1
```

```
*CONTROL_SHELL
20.0, , 1, -1, 1, 2, 2, 1
```

```
*CONTROL_CONTACT
1.0 0.0 2 2 1 1 1 0
0 0 10 0 4.0 1 0 0
```

Для каждого варианта расчёта найти главные деформации вблизи центральной части образца (в шейке), ориентироваться на отмеченные элементы на рис. 14 в):

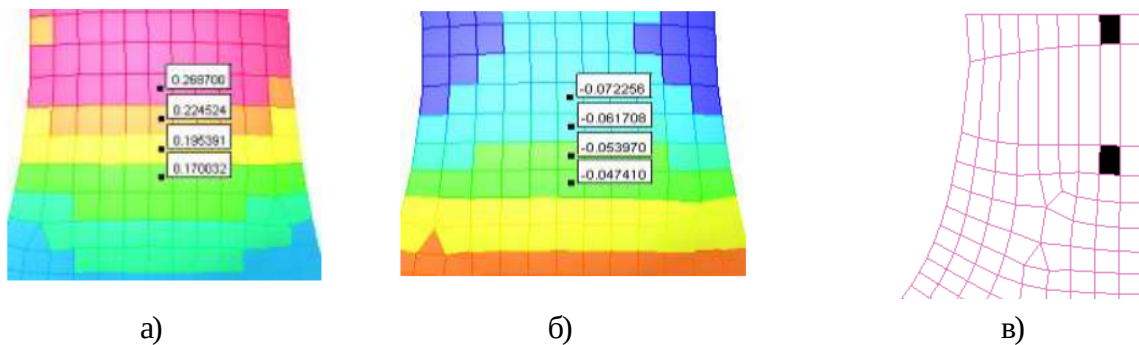


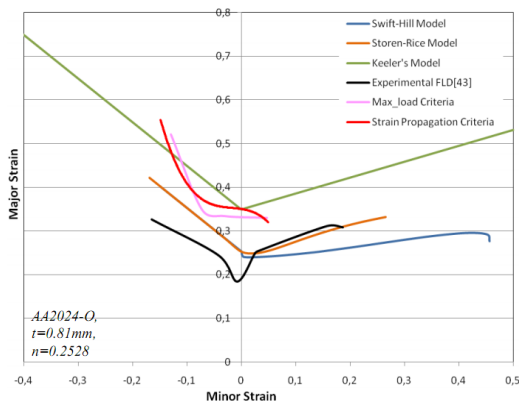
Рис. 14. КЭ-сетка: а) поле ϵ_1 (major strains) в момент потери устойчивости, б) поле ϵ_2 (minor strains) в момент потери устойчивости [58]; в) КЭ-сетка после потери устойчивости.

Значения главных деформаций занести в таблицу по шаблону в табл. 5 [58]; в шапку таблицы вписать обозначение материала, выбрать ваш вариант тест (M, H, X) и переменную, отвечающую за геометрию (h, b, r):

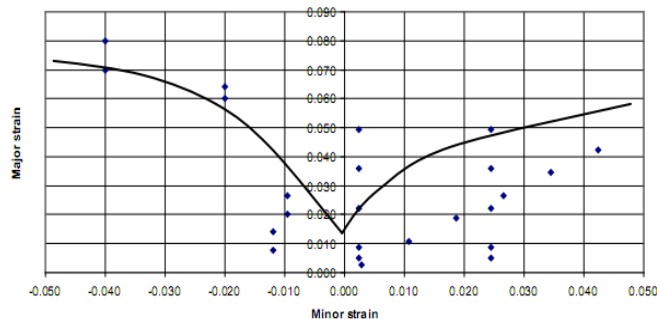
Таблица 5. Шаблон для записи результатов расчётов

Метод M/H/X	Результаты численного расчёта / материал _____			
	Толщина листа 1.2 мм		Толщина листа 2.5 мм	
h/b/r, мм	ϵ_2	ϵ_1	ϵ_2	ϵ_1
85	-0.0617	0.2245		
75	-0.038	0.2238		
65	-0.0060	0.2277		
60	0.0072	0.2277		
50	0.0122	0.1767		
40	0.0542	0.2222		
0	0.1718	0.2319		

Построить FLD на основе полученных табличных данных для двух вариантов толщин листа (1.2, 2.5 мм) и двух материалов, сравнить полученные кривые с известными на рис. 15.



а)



б)

Рис. 15. Примеры FLD: а) AA2024-O, толщина 0.81 мм; б) AA5182-O, толщина 1 мм . Источники: [41, 59]

Итоговое число расчётов на каждый вид теста: $2 \cdot 2 \cdot (8 \pm 1) = 32 \pm 4$. Для сокращения временных затрат автоматизировать процесс расчёта множества вариантов с помощью одного или нескольких *.bat-файлов.

Часть третья

Дело лишь в том, что вслед за вами пойдут другие... десятки других, сотни других.

Аркадий и Борис Стругацкие. Страна багровых туч.

4. Создание нового материала в LS-DYNA

4.1. Введение

Для внедрения нового материала в LS-DYNA требуется внести в исходники решателя программный код и пересобрать бинарник LS-DYNA. Большая часть исходников заранее скомпилирована в статическую библиотеку `ls-dyna.lib`, которая собирается под конкретную версию (971r4.2/r5/r5.1.1/r7 ...), платформу (32bit/64bit, SMP/MPP), операционную систему и предоставляется официальной техподдержкой Livermore Software Technology Corporation по запросу на сайте <http://www.lstc.com> (или их региональными дистрибьюторами). При положительном ответе, техподдержка пересылает в архиве необходимые исходники и прекомпилированные библиотеки^{11,12}.

Исходники написаны на языке FORTRAN 77 и 90. **Каждая версия LS-DYNA требует свою определённую версию компилятора** (например, решатели v971r7 и v971r9 собираются разными компиляторами). Основной компилятор под ОС Windows – Intel(R) Fortran Compiler (IFC, ifort)¹³. Версии LS-DYNA для Linux собираются и с помощью IFC, и с помощью GNU-того `gfortran`.

Для удобства разработки под ОС Windows **до установки IFC** устанавливаются: а) Microsoft Development Kit¹⁴; выбор версии MDK зависит от версии компилятора, номера рабочей ОС и года выпуска прекомпилированных библиотек LS-DYNA); б) среда разработки Microsoft Visual Studio C++, свободно распространяемая версия которой в первой половине 2000-х имела в названии приставку Express, а сегодня представлена версией Community¹⁵.

В качестве редактора исходного кода обычно используется функциональный свободно распространяемый Notepad++.

Например, для сборки 32-bit версии LS-DYNA `ls971r5.1.1` потребуются [20]:

- наличие локальной или сетевой лицензии LS-DYNA;
- компилятор Intel(R) Fortran Compiler for 32-bit applications v10.1;
- среда разработки Microsoft Visual C++ 2005 Express;
- исходники `ls971_s_R5.1.1_win32_p_lib` (по запросу у техподдержки).

Примечание: в релизах LS-DYNA R9 с 2017 года включительно разработчики реализовали возможность подключения к солверу независимых динамических библиотек с подпрограммами пользователя; известное ограничение LS-DYNA на 10 материалов пользователя (User Material, UM) в одном бинарнике теперь может быть устранено посредством корректировки процесса сборки бинарника и библиотек UM; добавились новые ключевые слова

11 Несколько релизов LS-DYNA (+ упомянутые архивы) доступны для свободного скачивания: http://www.oasys-software.com/dyna/en/downloads/ls-dyna_previous.shtml

12 Полный архив всех версий LS-DYNA, доступ по логину и паролю <http://www.lstc.com/download/ls-dyna>

13 IFC бесплатен для студентов, преподавателей, разработчиков открытого и свободного ПО, см. <https://software.intel.com/en-us/qualify-for-free-software>. Коммерческая 31-дневная триальная версия <https://software.intel.com/en-us/intel-parallel-studio-xe/try-buy/#parallelstudioxe>

14 Подробнее https://ru.wikipedia.org/wiki/Platform_SDK, https://en.wikipedia.org/wiki/Microsoft_Windows_SDK

15 Подробнее <https://www.visualstudio.com/ru/vs/cplusplus/>

для подключения данных библиотек в ходе выполнения расчёта. Подробности доступны в [33] и в документации релиза R9.

4.2. Этапы сборки бинарника с новым материалом (единый файл)

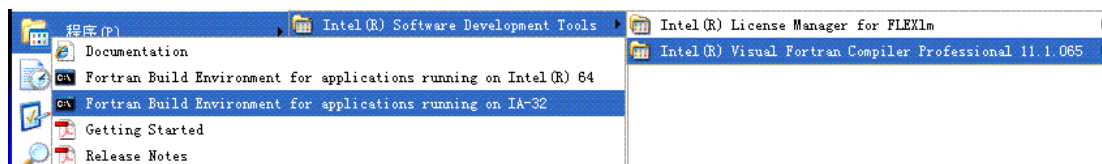
1) Запросите у техподдержки доступ к FTP-серверу ftp.lstc.com, скачайте с сервера архив с исходниками и библиотеками для приобретённой Вами версии LS-DYNA. Распакуйте архив в папку поближе к корню рабочего диска, например F:\r511win32. Данный архив, как правило, содержит:

- исходные коды на FORTRAN (*.f), в том числе основной исходник dyn21.f;
- заголовочные файлы (*.h);
- подключаемые исходники (*.inc);
- файлы динамических библиотек (*.so или *.dll);
- файлы статических библиотек (*.a или *.lib);
- скомпилированные файлы исходников, подготовленные для линковки (*.obj или *.o);
- файл сценария автосборки бинарника (Makefile);
- автосборщик (nmake.exe).

2) Откройте "dyn21.f" в Notepad++. Сформулируйте UM-код и внесите его в одну из процедур umatN(...) файла dyn21.f между комментария на рис. 3 "Начало кода UM" и "Конец кода UM".

3) Скомпилируйте исходник LS-DYNA и получите бинарник (новый решатель):

- под ОС Windows найдите в меню Пуск командную строку среды IFORТ "Visual Fortran Build Environment for IA-32 applications" (в случае 32-bit ОС и CPU) [31];



- перейдите в каталог с исходниками LS-DYNA:
 - команда смены каталога: cd .. и Enter или cd <path> и Enter;
 - смена диска: F: и Enter;


```

Intel(R) Visual Fortran Compiler Professional for applications runnin...
Intel(R) Visual Fortran Compiler Professional for applications running on IA-32,
Version 11.1.065
Copyright (C) 1985-2010 Intel Corporation. All rights reserved.

Setting environment for using Microsoft Visual Studio 2008 x86 tools.

C:\Documents and Settings\Administrator>cd c:\
C:\>f:
F:\>

```

```

Intel(R) Visual Fortran Compiler Professional for applications runnin...
Intel(R) Visual Fortran Compiler Professional for applications running on IA-32,
Version 11.1.065
Copyright (C) 1985-2010 Intel Corporation. All rights reserved.

Setting environment for using Microsoft Visual Studio 2008 x86 tools.

C:\Documents and Settings\Administrator>cd c:\
C:\>f:
F:\>cd r511win32
F:\r511win32>nmake_

```

- наберите nmake (make для Linux) и Enter для компиляции UM:

```

Intel(R) Visual Fortran Compiler Professional for applications runnin...
Setting environment for using Microsoft Visual Studio 2008 x86 tools.

C:\Documents and Settings\Administrator>cd c:\
C:\>f:
F:\>cd r511win32
F:\r511win32>nmake

Microsoft (R) Program Maintenance Utility   Version 6.00.8168.0
Copyright (C) Microsoft Corp 1988-1998. All rights reserved.

        ifort -c -W0 -WB -unroll -fp:precise -4Yportlib -assume:byterecl,buffered
d_io -Qfpp2 -DPCWIN -DINTEL -QxW -DIA32 -DIA32ONLY -DLSTCODE -DLONGFORMAT -DOPE
NMP -Qopenmp -DDOUBLEXR dyn21.F
Intel(R) Visual Fortran Compiler Professional for applications running on IA-32,
Version 11.1   Build 20100414 Package ID: w_cprof_p_11.1.065
Copyright (C) 1985-2010 Intel Corporation. All rights reserved.
ifort: command line warning #10212: /fp:precise evaluates in source precision wi
th Fortran.

```

Генерация бинарника займёт некоторое время. Будет создан солвер LS-DYNA, например, с именем ls971.exe:

```

c:\ Intel(R) Visual Fortran Compiler Professional for applications runnin...
libansys.lib(lm_new.obj):warning : locally defined symbol __imp__time imported
Microsoft (R) Incremental Linker Version 9.00.21022.08
Copyright (C) Microsoft Corporation. All rights reserved.

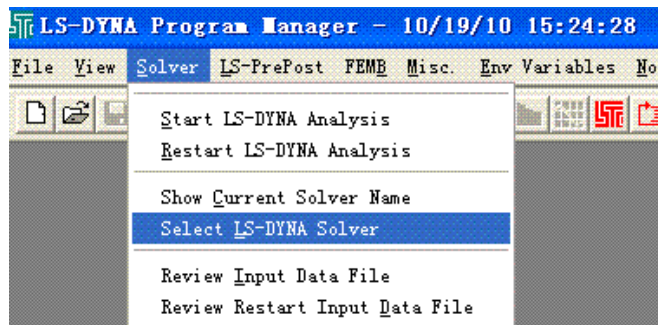
-out:ls971.exe
-subsystem:console
-stack:4000000
-force
-nodefaultlib:vcomp.lib
-nodefaultlib:msvcrt.lib
dyn21.obj
dyn21b.obj
libdyna.lib
libansys.lib
libiomp5mt.lib
shell32.lib
user32.lib
comctl32.lib
comdlg32.lib
Iphlpapi.lib
libansys.lib(lm_new.obj) : warning LNK4217: locally defined symbol _time importe
d in function _l_buf_21
F:\r511win32>
    
```

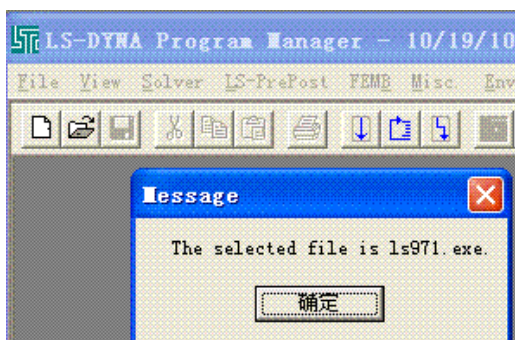
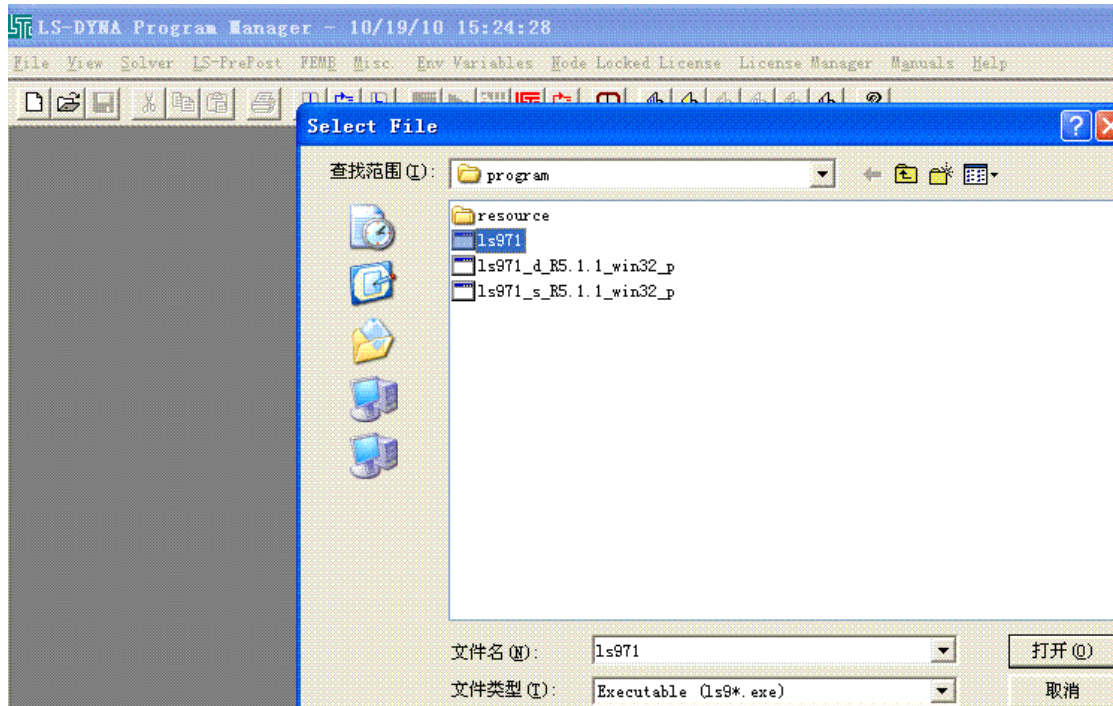
sorter_noioc	4 KB	include file
systeminfo	1 KB	Include File
txtline	1 KB	Include File
umatss	1 KB	Include File
xjobid	1 KB	Include File
dyn21	244 KB	Object File
dyn21b	711 KB	Object File
ls971	985 KB	应用程序

4) Скопируйте созданный бинарник ls971.exe в основную папку пакета LS-DYNA, например C:\LSDYNA\program.

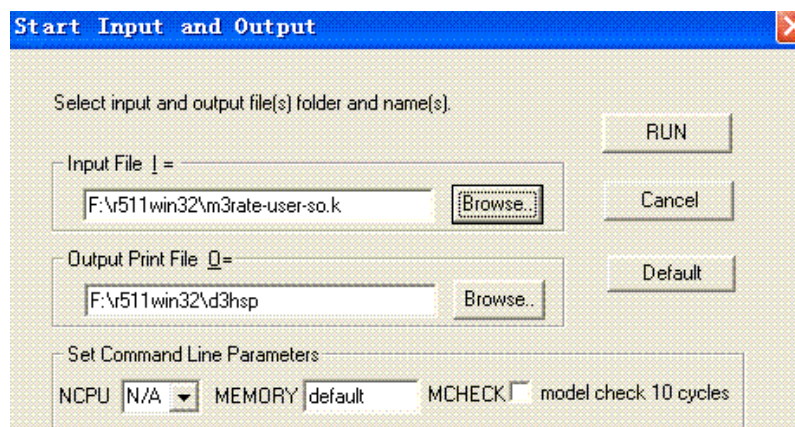
5) Подключите созданный материал (или иной созданный функционал) к расчётному к-файлу. Новый материал подключается ключевым словом *MAT_USER_DEFINED_MATERIAL_MODELS

6) В настройках LS-DYNA Manager выберите новый решатель:





7) Далее k-файл отправляется на расчёт в обычной последовательности:



```

LS-DYNA Program Manager - 10/19/10 15:24:28
File View Solver LS-PrePost FEMM Misc. Env Variables Node Locked License License Manager Manuals Help
C:\LS-DYNA511\program\ls971.exe I=F:\r511win32\m3rate-user-so.k O=F:\r511win32\d3hsp
ASCII database ..... 0.0000E+00 0.00 3.1000E-02 0.53
Contact algorithm .... 0.0000E+00 0.00 4.8000E-02 0.82
Contact entities ..... 0.0000E+00 0.00 0.0000E+00 0.00
Rigid bodies ..... 0.0000E+00 0.00 1.6000E-02 0.27
Implicit Nonlinear ... 0.0000E+00 0.00 0.0000E+00 0.00
Implicit Lin. Alg. ... 0.0000E+00 0.00 0.0000E+00 0.00
-----
T o t a l s          6.0000E+00 100.00 5.8280E+00 100.00

Problem time       = 1.0000E-03
Problem cycle      = 7935
Total CPU time     = 6 seconds < 0 hours 0 minutes 6 seconds>
CPU time per zone cycle = 1120 nanoseconds
Clock time per zone cycle= 1212 nanoseconds

Number of CPU's   1
NLQ used/max      96/ 96
Start time        07/22/2011 12:55:34
End time          07/22/2011 12:55:40
Elapsed time      6 seconds< 0 hours 0 min. 6 sec.> for 7935 cycles

Normal termination                                07/22/11 12:55:40

```

4.3. Примеры скриптов автосборки (Makefile)

Общая методика работы с Makefile и принципы построения скриптов доступно изложены в документации к утилите make. Не смотря на то, что в стандартный Makefile обычно не требуется вносить изменения, приведём несколько частных случаев скрипта.

1) Для компилятора Intel Fortran, среда Intel XE 2013 (Linux)

```

ls971: dyn21.o dyn21b.o couple2other_user.o
  ifort -o ls971 init_once.o init_keywrd.o dynm.o dyn21.o dyn21b.o
couple2other_user.o adummy_graph.o orderByMetis.o lcfunc.o adummy_fftw3.o
adummy_msc.o mscapi.o libdyna.a libbcsext4.a liblsda.a liblssecurity.a libmetis.a
liblcpack.a libspooles.a libcparse.a libmf2.a liblsm.a liblscrypt.a libresurf.a
intel64_101_libansys.a libdyna.a -i-static -L/usr/X11R6/lib64 -lX11 -openmp -
lstdc++ -lrt
dyn21.o: dyn21.f nhisparm.inc
  ifort -c -w95 -zero -assume byterecl,buffered_io,protect_parens -
mP2OPT_hpo_dist_factor=21 -ftz -nopad -fpp2 -openmp -xW -fp-model precise -O2 -I.
dyn21.f
dyn21b.o: dyn21b.f nhisparm.inc
  ifort -c -w95 -zero -assume byterecl,buffered_io,protect_parens -
mP2OPT_hpo_dist_factor=21 -ftz -nopad -fpp2 -openmp -xW -fp-model precise -O2 -I.
dyn21b.f

```

2) Для компилятора gfortran (Linux)

```

lsdyna: dyn21.o dyn21b.o couple2other_user.o
  /usr/bin/gfortran -o lsdyna init_once.o init_keywrd.o dynm.o dyn21.o dyn21b.o
couple2other_user.o adummy_graph.o orderByMetis.o adummy_msc.o mscapi.o
libdyna.a libbcsext4.a liblsda.a liblssecurity.a liblcpack.a libspooles.a
libcparse.a libmf2.a liblsm.a liblscrypt.a libresurf.a libsfg.a libmetis.a
libfemster_wrap.a libfemster_wrap2d.a libfemster_wrap1d.a libfemster.a
libfemster2d.a libfemster1d.a libpfem.a libmetis.a libblas.a liblapack.a
libfftw3.a libsprng.a intel64_101_libansysdp.a libdyna.a libblas.a liblapack.a

```

```

libmetis.a -i-static -L/usr/X11R6/lib64 -lX11 -openmp -lstdc++ -lrt -lstdc++
dyn21.o: dyn21.f nhisparm.inc
    /usr/bin/gfortran -c -w95 -zero -assume byterecl,buffered_io,protect_parens -
mP2OPT_hpo_dist_factor=21 -ftz -nopad -fpp2 -openmp -i8 -r8 -DINTEL -DAdd_ -xW -
fp-model precise -O2 -I. dyn21.f
dyn21b.o: dyn21b.f nhisparm.inc
    /usr/bin/gfortran -c -w95 -zero -assume byterecl,buffered_io,protect_parens -
mP2OPT_hpo_dist_factor=21 -ftz -nopad -fpp2 -openmp -i8 -r8 -DINTEL -DAdd_ -xW -
fp-model precise -O2 -I. dyn21b.f
couple2other user.o: couple2other_user.f
    /usr/bin/gfortran -c -w95 -zero -assume byterecl,buffered_io,protect_parens -
mP2OPT_hpo_dist_factor=21 -ftz -nopad -fpp2 -openmp -i8 -r8 -DINTEL -DAdd_ -xW -
fp-model precise -O2 -I. couple2other_user.f

```

3) Сборка динамической библиотеки материалов (Linux) [71]

```

MODEL = NF
OBJS = $(MODEL).o dyn21.o dyn21b.o
OPTIONS = -c -w95 -W0 -zero -safe_cray_ptr -assume byterecl,buffered_io -
mP2OPT_hlo_fusion = F -save -traceback -save -pad -nodps -DLINUX -
DNET_SECURITY -DADDR64 -DINTEL -DXEON64 -DFCC80 -DMPP -DMPICH -
DHPMPI -DAUTODOUBLE -DNEWIO -i8 -r8 -xW -fpic -O2 -I.
mpp971: $(OBJS)
    mpif90 -shared -o libmpp971_d_7600.2.1224_usermat.so $(OBJS)
$(MODEL).o: $(MODEL).f
    mpif90 $(OPTIONS) $(MODEL).f
dyn21.o: dyn21.f
    mpif90 $(OPTIONS) dyn21.f
dyn21b.o: dyn21b.f
    mpif90 $(OPTIONS) dyn21b.f

```

4) Линковка стороннего объектника [32]

```

OBJS = dyn21.obj dyn21b.obj umat41-for-shell.obj
FC = ifort
FFLAGS = -O2 -I. -L.

ls971: dyn21.o dyn21b.o umat41-for-shell.obj
    $(FC) $(FFLAGS) -o ls971 $(OBJS) intel64_101_libansys.a libdyna.a
dyn21.obj: dyn21.F
    $(FC) $(FFLAGS) dyn21.F
dyn21b.obj: dyn21b.F
    $(FC) $(FFLAGS) dyn21b.F
umat41-for-shell.obj: umat41-for-shell.f
    $(FC) $(FFLAGS) umat41-for-shell.f

```

4.4. Дополнения к коду решателя. User Subroutines

В сущности, солвер LS-DYNA есть не что иное, как выверенные за десятилетия математические алгоритмы для обработки поведения различных типов элементов, материалов, контактов, соединений, нагрузок, граничных условий и т.д. Алгоритмы постепенно совершенствуются, вносятся изменения качественного характера, исправляются выявленные ошибки.

Стандартные алгоритмы могут не подходить исследователю в силу каких-либо ограничений: не устраивают возможности известных моделей поведения металлов, таких, как пластики, эластики, пены, бетоны, почвы, композиты, дерево и пр.; известные изотропные упругие и анизотропные упруго-пластические не учитывают характерные особенности повреждения и т.д. Для подобных случаев LS-DYNA позволяет дополнить собственный код силами пользователя, тем самым, предоставляя исследователю некоторую свободу выбора метода решения задачи [21, 24-27].

В табл. 6 представлен доступный для перепрограммирования функционал LS-DYNA (подробнее [4, 25]). Чаще всего дописываются модели поведения материалов¹⁶. В один бинарник (исполняемый файл) возможно включить до 10 различных новых моделей материалов (при этом количество бинарников не ограничено; также см. примечание в пункте 4.1).

Таблица 6. User-Defined Features & corresponding keyword input, subroutines in the user interface and files

Модифицируемый функционал		Ключевое слово	Имя функции	Файл с функцией
Structural Mechanics		*MAT_USER_DEFINED...	usermat, utans	dyn21.f
Equation of state	Уравнение состояния	*EOS_USER_DEFINED	ueoslib	dyn21b.f
Cohesive materials		*MAT_USER_DEFINED	umatXXc	dyn21b.f
Thermal materials	Теплопроводность	*MAT_THERMAL_USER ...	thusmat	dyn21b.f
Material failure e.g.	Критерий разрушения	*MAT_024 with FAIL<0	matusr_24/103	dyn21.f
Structural elements	Конечные элементы	*SECTION SHELL/SOLID with ELFORM=101...105	usrshl, usrsld	dyn21b.f
Interface friction	Функционал трения	*USER_INTERFACE_CONTROL	usrfric	dyn21.f
Solution (output) control	Контроль процесса решения и сохранения результатов	-	uctrl1	dyn21.f
Interface control	Функционал контроля	*USER_INTERFACE_CONTROL	uctrl2	dyn21.f
Airbag sensor	Сенсор подушки безопасности	*AIRBAG: parameter RBID	airusr	dyn21.f
Thermal contact	Термический контакт	*USER_INTERFACE_CONDUCT ...	usrhcon	dyn21.f
Boundary flux	ГУ: поток	*BOUNDARY FLUX: NHISV>0	usrflux	dyn21.f
Adaptively	Адаптивность	*CONTROL_ADAPTIVE with ADPOPT=9	useradap	dyn21.f
Loads	Нагрузки	*USER_LOADING	loadud	dyn21.f
Weld failure	Разрушение сварного соединения	*MAT_100 with OPT=2,12,22	uweldfail	dyn21.f
Joint forces	Реакции в шарнирах	*CONSTRAINED_JOINT_USER	ujntdrv	dyn21.f

16 См. http://www.dynalook.com/search?SearchableText=umat&submit=Search&portal_type%3Alist=File и http://www.dynalook.com/search?SearchableText=subroutine&submit=Search&portal_type%3Alist=File

4.5. Новый материал. User Material (UM, UMAT).

Квалифицированному исследователю предоставлена возможность описать поведение нестандартных материалов своими собственными моделями. Процесс создания нового материала разбивается на этапы (см. [75-81]):

- выписать предполагаемые физические уравнения^{17, 18} (constitutive model), которые свяжут напряжения и деформации;
- переложить теоретическую модель на язык математики: выписать уравнения в компактной форме;
- задать функцию течения (flow stress function)¹⁹ (или диаграмму деформирования - отношение между истинными напряжениями и истинными (логарифмическими) деформациями (flow curve), пользуясь гипотезой единой кривой);
- выбрать поверхность текучести (например, по Мизесу для случая ПНС);
- учесть параметры, влияющие на изменение напряжений текучести (температура, скорость деформирования, повреждённость и т.д.);
- выбрать схему интегрирования (например, по напряжениям);
- оформить теоретическую модель в виде подпрограммы UM на языке FORTRAN; внести код подпрограммы в файл dyn21.f; скомпилировать рабочий бинарник LS-DYNA (.exe);
- провести натурные эксперименты с образцами;
- выполнить верификацию UM, сравнив поведение натуральных образцов с результатами вычислительных экспериментов; успешная верификация послужит гарантией работоспособности алгоритма и правильности разработанной теоретической модели материала.

Исходные коды пользовательского материала именуются подпрограммами umat41(...), umat42(...), ..., umat50(...) (для неявного решателя utanN()), которые, обыкновенно, содержатся в файле dyn21.f. На рис. 16 представлена схема обмена данными между солвером и UM.

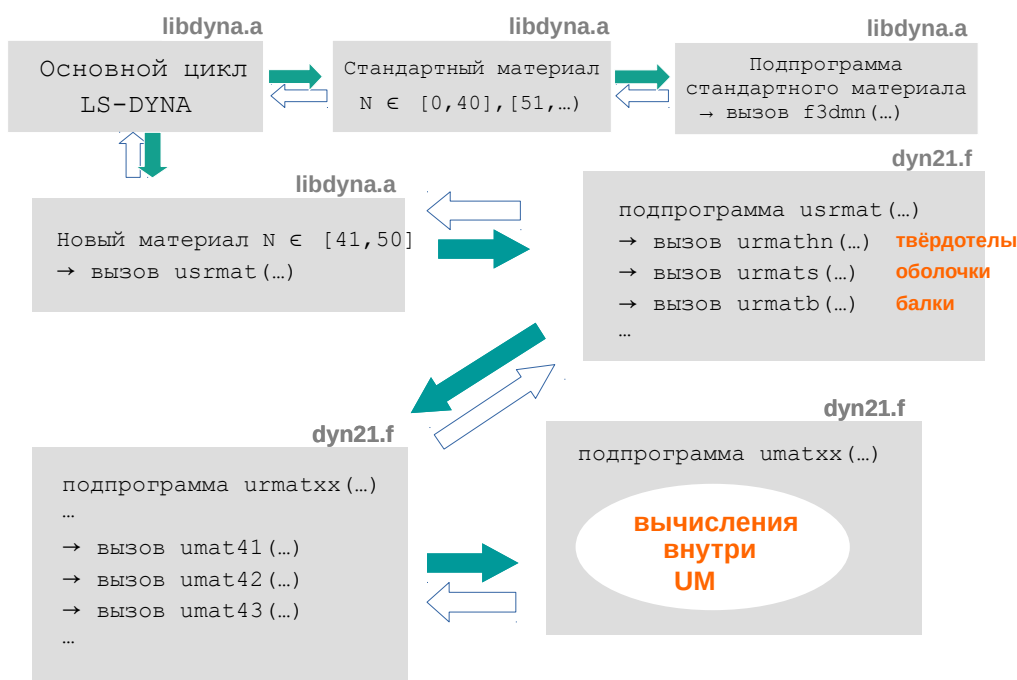


Рис. 16. Иерархия и порядок вызова функций. Стрелки показывают обмен данными между подпрограммами. Источники: [25, 27]

17 Уравнения состояния описывают нелинейные соотношения между действительными напряжениями, действительными скоростями деформации и температурой при различном уровне деформаций [34 С. 904].

18 Простейший пример см. например [82 С. 67]

19 Flow stress == true stress == effective stress [34 С. 906],

На каждом временном шаге для каждого элемента элементный обработчик солвера вызывает из файла `dyn21.f` процедуру `usrmat(...)`, а затем, в зависимости от типа и свойств элемента, вызываются подпрограммы `umatN(...)` и в них выполняются те или иные блоки кода, в которых вычисляются приращения деформаций, находятся напряжения по приращениям деформаций, проводится проверка повреждаемости, изменяются переменные истории.

Примечание. В архиве с исходниками чаще всего подвергаются редактированию файлы `dyn21.f` и `dyn21b.f`. Данные файлы содержат в себе множество подпрограмм, все они могут вызываться элементным обработчиком LS-DYNA. Для повышения эффективности разработки исходник разбивается на отдельные файлы. Связать разделённые подпрограммы возможно с помощью редактирования скрипта `makefile` (см. Линковка стороннего объектника [32]), или подключив отдельные подпрограммы из внешних файлов директивой `include`. Например, выделим в отдельный файл подпрограмму `umat41(...)`, для чего вырезаем полный текст подпрограммы `umat41(...)` из исходника `dyn21.f`, сохраняем его в файл `umat41.f` и подключаем файл в основном исходнике `dyn21.f`:

```
!  
include 'umat41.f'
```

По целевому назначению подпрограммы файла `dyn21.f` делятся на явные и неявные (для соответствующих решателей). Явные подпрограммы вызываются после выполнения шага, например `umat41` вычисляет напряжения по приращению деформаций и изменению температуры. Неявные подпрограммы вызываются до выполнения шага, к примеру, `utan42` меняет модуль Юнга в элементе, или `utan41`, которая изменяет элементную матрицу жёсткости (в итоге и глобальную).

4.6. Базовый синтаксис FORTRAN и пример UM

При написании UM для LS-DYNA v971 используется современный синтаксис FORTRAN-95/2003/2008, по своей сути мало отличающийся от знакомого инженерам-математикам синтаксиса Matlab/Scilab. Основное отличие заключено в явном указании типа переменных. Изначально исходные тексты рабочих примеров UM писались на FORTRAN-77/90, поэтому для понимания работы примеров (они, как правило, не переписываются на современный диалект) требуется пояснение структуры строк.

В FORTRAN-77 принят фиксированный формат строк²⁰: операторы программы располагаются с 7-й или далее по 72-ю колонку строки (см. рис. 17): в первых пяти колонках строки ставится метка для операторов `do` и `goto`, в шестой колонке ставится знак переноса – любой символ, обычно `"*"`, наличие которого говорит о том, что данная строка является продолжением предыдущей; комментарии пишутся после символа `"c"` или `"*"`, стоящего в первой колонке строки, а также после восклицательного знака (!), который можно ставить в любой колонке строки. В FORTRAN-90 используется как фиксированный формат, так и свободный²¹. Синтаксис FORTRAN рассмотрим далее на части кода подпрограммы `umat41(...)`, также см. Исходные тексты UM на языке FORTRAN.

20 Краткий курс FORTRAN http://malex13.narod.ru/f/f_ind.htm

21 См. например [Акимова Е.Н. Основы программирования на языке FORTRAN](#)



Рис. 17. Фиксированная структура строки FORTRAN-77²²

```
! В ПЕРВЫХ СТРОКАХ УКАЗЫВАЕМ АВТОРОВ КОДА И ДАТУ ВНЕСЕНИЯ ИЗМЕНЕНИЙ
! Авторы: Петрова А.А., Сидоров Б.Б.
! Дата : 01 сентября 2017 версия 1.1
! e-mail: petrova_i_sidorov@mail.ru
! Тел. : +7 000 000 00 00
! ООО Р.и.К. Все права защищены
!
! 6-я позиция
! 8-я позиция
! 73-я позиция !
! исходный текст подпрограммы
subroutine umat41 (cm,eps,sig,epsp,hsv,dtl,сара,etype,tt,
1 temper,failel,сrv,сma,qmat,elsiz,idele)
! подключаем внешний код, макросы
include 'nlqparm'
include 'iounits.inc'
! определяем структуры и массивы
common/bk06/idmmy,iaddp,ifil,maxsiz,ncycle,time(2,30)
dimension cm(*),eps(*),sig(*),hsv(*),сrv(101,2,*)
! действительные переменные
real dtl,сара,tt
! строковые переменные
!character*(*) etype !без задания длины
character*5 etype
! логические переменные
logical failel
! вывод сообщений в консоль по условию
! начало исполнительного блока
if (ncycle.eq.1) then ! if (ncycle == 1) then
call usermsg('mat41') ! call usermsg('mat41T')
write(*,*)
write(*,*) 'Predelnye napryazheniya: ', cm(5)
write(*,*)
endif
! далее размещается основной код
! КОД UM
! разрушение по заданному уровню эквив. напряжений Мизеса
sig_eqv=sqrt(((sig(1)-sig(2))**2+(sig(2)-sig(3))**2+
& (sig(3)-sig(1))**2+6*(sig(4)**2+sig(5)**2+sig(6)**2))/2)
if (sig_eqv.gt.cm(5)) failel=.true.
! экстренная остановка расчёта по условию
if (failel.eq..true.) then
STOP
endif
! выход из подпрограммы, окончание текста подпрограммы
return
end
```

22 Источник: Михайленко К.И. FORTRAN-77: базовые понятия. <http://const.anrb.ru/bashgu/cs/fortran>

4.7. Подключение UM в расчётном файле

Ключевое слово `*MAT_USER_DEFINED_MATERIAL_MODELS` подключает новый материал (подробнее см. Keyword Manual для Вашей версии LS-DYNA):

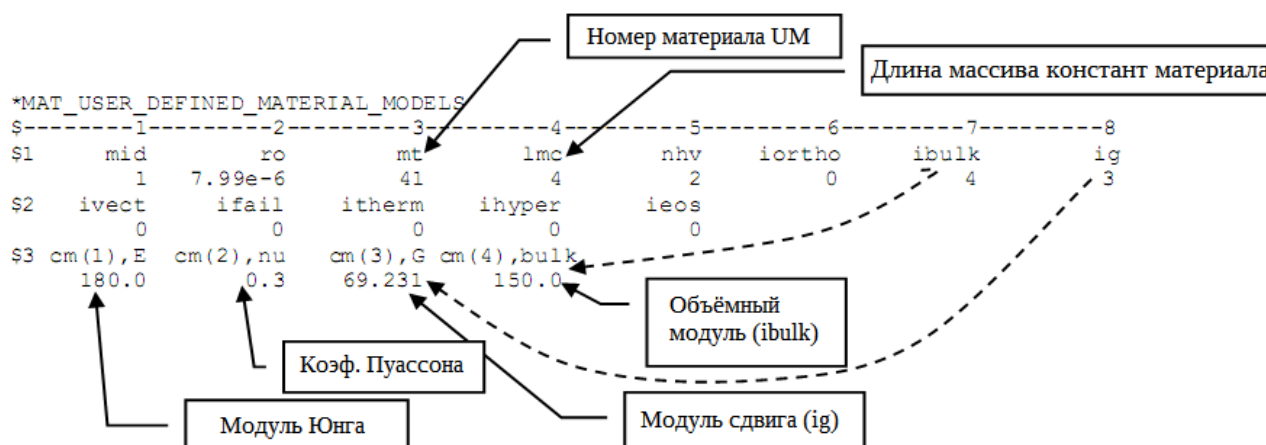


Рис. 18. Подключение UM в расчётном файле. Источник: [21]

`mid` – идентификатор материала в модели,

`ro` – плотность,

`mt` – номер UM в исходниках, принимает значение от 41 до 50,

`lmc` – длина массива констант материала `cm(i)`, до 48 включительно,

`lmsa` – длина дополнительного массива констант материала `cm(i)`,

`nhv` – количество сохраняемых переменных истории,

`iortho = 1` – учитывать орто/анизотропию свойств,

`ibulk` и `ig` – позиция значений объёмного модуля и модуля сдвига в массиве параметров `cm(i)`,

`ivect` – флаг использования векторизованных пользовательских подпрограмм `umatNv`,

`ifail` – флаг разрушения, если не 0 – разрушение разрешено для оболочек и твёрдотельников,

`itherm` – учёт температуры; если =1, то в переменной `temper` содержится температура (на любом шаге в связанном термомеханическом анализе тепловая задача решается первой),

`ihyper`, `ieos` и др. – см. Keyword Manual.

4.8. Массивы и переменные в UM. Баги, недокументированные особенности, информация к сведению

Внимательно изучите Keyword Manual Appendix A! (и последующие приложения)

1) Процедуры UM делятся на скалярные и векторные²³. Скалярные предназначены для последовательного выполнения на CPU – одна за другой по мере перебора конечных элементов, векторные выполняются параллельно с помощью векторных инструкций (MMX, SSE, AVX для процессоров Intel). Перечисленные в таблице 7 структуры данных предназначены для скалярной процедуры UM, они будут встречаться пользователю чаще всего.

Перечисленные в табл. 8 переменные есть векторизованная форма переменных из табл. 7, здесь `nlq` – длина массива точек интегрирования КЭ-сетки. Для обработки в массиве доступны позиции от `llt` до `lft` (на входе в подпрограммы данные переменные заполняются солвером).

Текущая скорость деформирования получается делением приращения деформации `eps` на текущий шаг по времени `dt1`. Напряжения `sig` с прошлого шага передаются в подпрограмму

²³ См. историю создания суперкомпьютера Cray-1, для работы на котором Ливерморской национальной лабораторией Министерства энергетики США были векторизованы исходные коды DYNA2D и DYNA3D.

пользователя, затем в ней обновляют свои значения, и, в итоге, возвращается из umat как суммарные напряжения в момент времени t+dt.

Таблица 7. Структуры данных и аргументы скалярных пользовательских подпрограмм [23]

Переменная	Описание
cm(1)...cm(N)	N-я константа материала, до 48;
eps(1)...eps(6)	Приращение деформации в локальной системе координат по x, y, z, xy, yz, zx
epsr	Результирующая (effective) пластическая деформация на предыдущем шаге; добавлена в 971-й
sig(1)...sig(6)	Напряжения в ЛСК по x, y, z, xy, yz, zx (in the previous time)
hsv(1)...hsv(n)	N-я переменная истории на предыдущем шаге, исключая пластическую деформацию; на первой итерации = 0; массив именовался hisv(n) до релиза 971-й LS-DYNA
capa	Коэффициент коррекции поперечного сдвига (transverse shear correction or reduction factor)
dt1	Текущий шаг по времени
etype	= "solid" (ранее "brick") для твёрдотельных элементов, "shell" - оболочка (формулировка 1 и 2), "beam" - балка, "sld2d" - плоские элементы (13, 14, 15), "tshel" - плиты (3, 5), "hbeam" - балки (1, 11), "tbeam" - стержень (3), "dbeam" - пружинка/демпфер (6), "shl_t" - оболочка (25,26,27)
fail1	Присвойте флагу .true. чтобы разрушить элемент; если уже установлен в .true., то элемент разрушен ранее
tt	Текущее время в расчёте
temper	Текущая температура
crv(101,2,*)	Массив некоторой табличной функции, заданной в расчётном файле (подробнее [36])
cm(*)	Массив дополнительных констант материала (см. lmsa выше)
elsiz	Характеристика размера элемента
idele	Идентификатор элемента
es(6,6)	Матрица упругих констант в подпрограммах utan

Таблица 8. Векторизованные структуры данных в LS-DYNA [23]

Структура данных	Описание
cm(*)	Весь массив констант (в векторных подпрограммах usermatNv)
d1(nlq)...d6(nlq)	Приращения деформаций
epsps(nlq)	Результирующая пластическая деформация на предыдущем шаге
sig1(nlq)...sig6(nlq)	Вектор напряжений на предыдущем шаге
hsvs(nlq,*)	Переменные истории на предыдущем шаге за исключением пластической деформации
failels(nlq)	Массив флагов разрушения, позиция в массиве указывает номер элемента
dt1siz(nlq)	Текущий шаг по времени
temps(nlq)	Текущая температура

Векторная запись UM соответствует нотации Кельвина-Фойгта, например, вектор напряжений имеет следующий вид [24]:

$$[sig(1:6)]^T = [\sigma_{11} \sigma_{22} \sigma_{33} \sigma_{32} \sigma_{31} \sigma_{12}] \cdot \quad (16)$$

2) eps(1)...eps(6) – это приращения **инженерной** деформации [24, 36].

3) Вероятно, что первая сложность, с которой встречаются при программировании анизотропных моделей поведения материала, – это ограниченное количество задаваемых констант материала, которые солвер способен переслать из расчётного файла в UM, а также фиксированный размер некоторых других массивов [24].

Константы материала, как правило, содержатся в массиве CM. В пользовательскую процедуру допустимо переслать 40 констант для ортотропного материала и 50 для изотропного, из которых две последние позиции отводятся для объёмного модуля и модуля сдвига, т.о. доступно 38 и 48 позиций соответственно. Ранее ограничение обходили путём записи констант во внешний текстовый файл и дальнейшим чтением констант из внешнего файла в процедуре UM, или заданием констант непосредственно в исходнике процедуры UM [36] (причём, способ чтения из файла не лишён недостатка в SMP-среде, подробнее [23]):

```

    if (hist(1).lt.0.1) then
c      ...чтение констант из файла
        open(11, file='material.ini')
        do 1 i=1,cm(1)
            read (11,*) cm(i+1)
1      continue
        close (11)
c      ...завершение блока инициализации
        hist(1)=1
    endif

```

Когда расчётный файл содержит более 48 констант материала, все позиции массива CM после 48-го обнуляются безо всякого предупреждения со стороны солвера. В свежих релизах появилась возможность задания дополнительного массива констант CMAdditional.

4) В солвере 2001-го года количество переменных истории не должно превышать 100 (в релизах r4 и выше — 142 по-умолчанию, см. константу NHISVAR в файле nhisparm.inc), из них 6 первых отводятся под матрицу поворота. Т.е., в наличии 94 свободных позиции, и, если требуется выделить 94 переменные, то задайте nhv=100 [36]. Переменные истории доступны в каждом элементе (или гауссовой точке). Они могут использоваться в качестве идентификаторов, хранилища промежуточных данных, сумматоров пластической деформации и прочих переменных, которые пользователь пожелает сохранить между двумя вычислительными шагами по времени или использовать на последующих шагах в своих подпрограммах. На первом шаге массивы переменных истории обнуляются (как и пластические деформации).

Дополнительные N-переменных истории для всех типов КЭ сохраняются в файл d3plot, если добавить в расчётный k-файл ключевое слово

```

*DATABASE_EXTENT_BINARY
N,N,N,1

```

Данное ключевое слово приводит к сохранению N-переменных для всех материалов модели вне зависимости, используют ли они свои переменные истории или нет, что значительно увеличивает размер базы данных результатов на жёстком диске.

В 2007-м году был обнаружен следующий баг: если расчётный файл содержит несколько ключевых слов *MAT_USER_DEFINED_MATERIAL_MODELS, тогда параметр nhv первого материала используется для всех пользовательских материалов в модели. Это серьёзная проблема коснулась материалов, в которых используется деформационный градиент.

LS-DYNA позволяет сохранить переменные истории и в векторном блоке при установке флага ivect=1. Затем переменные могут быть сохранены в вектор-массиве длиной nlq с векторным индексом в диапазоне от lft до llt.

5) Следующий баг ls971 2007 года: с любым решателем (явным или неявным) когда вы используете UM, который вычисляет напряжения на основании деформационного градиента, при включении mass scaling в ключевом слове *CONTROL_TIMESTEP решатель падает на первом же шаге. Поэтому используйте обходной путь через изменение плотности материала в ключевом слове *MAT_USER_DEFINED_MATERIAL_MODELS.

6) Когда для работы модели поведения материала требуется деформационный градиент, то установка в ключевом слове *MAT_USER_DEFINED_MATERIAL_MODELS соответствующего флага приводит к сокращению доступных пользователю переменных на 9 позиций (в них солвер сохраняет значения деформационного градиента).

5. Примеры УМ

5.1. Линейно упругий материал

См. код примера Линейно-упругий материал. Реализуются линейные физических уравнениях:

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij}, \quad (17)$$

где σ_{ij} и ε_{ij} компоненты тензора напряжений и деформаций, δ_{ij} - дельта Кронекера, λ и μ - константы Ламэ. В таблице 9 поясняются основные расчётные формулы кода ЛУМ.

Таблица 9. Реверс-инжиниринг исходного кода примера 1 [28]

Вычисления для элементов типа SOLID		Вычисления для элементов типа SHELL	
Код	Физический смысл	Код	Физический смысл
$g2 = \text{abs}(cm(1)) / (1. + cm(2))$ $g = .5 * g2$	Модуль сдвига $G = \frac{E}{2(1+\nu)}$	-/-	-/-
		$gc = cava * g$	Сдвиговая коррекция $G_k = \kappa G$
		$q1 = \text{abs}(cm(1)) * cm(2) / ((1.0 + cm(2)) * (1.0 - 2.0 * cm(2)))$ $q3 = 1. / (q1 + g2)$	Параметры Ламе $\lambda = \frac{E \nu}{(1+\nu)(1-2\nu)}$; $\mu = \frac{1}{Q1 + 2G}$
		$eps(3) = -q1 * (eps(1) + eps(2)) * q3$	$\varepsilon_3 = \lambda \mu (\varepsilon_1 + \varepsilon_2)$
$davg = (-eps(1) - eps(2) - eps(3)) / 3.$	Средняя деформация (с отриц. знаком) $S = -\frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3}{3}$	-/-	-/-
$p = -davg * \text{abs}(cm(1)) / (1. - 2. * cm(2))$	Объёмный модуль и гидростатическое давление $K = \frac{E}{1-2\nu}; p = -SK$	-/-	-/-
$sig(1) = sig(1) + p + g2 * (eps(1) + davg)$ $sig(2) = sig(2) + p + g2 * (eps(2) + davg)$ $sig(3) = sig(3) + p + g2 * (eps(3) + davg)$	Диагональные компоненты тензора напряжений $\sigma_i^n = \sigma_i^{n-1} + p^n + 2G(\varepsilon_i^n - S)$	$sig(1) = sig(1) + p + g2 * (eps(1) + davg)$ $sig(2) = sig(2) + p + g2 * (eps(2) + davg)$ $sig(3) = 0.0$	-/-
$sig(4) = sig(4) + g * eps(4); sig(5) = sig(5) + g * eps(5); sig(6) = sig(6) + g * eps(6)$	Сдвиговые компоненты тензора напряжений $\sigma_{ij}^n = \sigma_{ij}^{n-1} + G \varepsilon_{ij}^n$	-/-	-/-

Напряжения на текущем временном шаге σ_{ij}^n получаются суммированием напряжений на предыдущем шаге σ_{ij}^{n-1} и приращений напряжений на текущем шаге, которые, в свою очередь, зависят от приращения деформаций $\Delta \varepsilon_{ij}^n$ на текущем шаге. Физические уравнения (17) переписываются для реализации шагово-итерационного метода:

$$\sigma_{ij}^n = \sigma_{ij}^{n-1} + \lambda \delta_{ij} \Delta \varepsilon_{kk}^n + 2\mu \Delta \varepsilon_{ij}^n. \quad (18)$$

Для рассматриваемого материала модуль Юнга задаётся константой $\sigma(1)$, коэффициент Пуассона константой $\sigma(2)$, для элементов оболочек потребуется задать значение коэффициента коррекции поперечного сдвига κ . Пример подключения ЛУМ к расчётному файлу см. на рис. 18.

5.2. Изотропный упруго-пластический материал

См. код примера Упруго-пластический материал по Мизесу и алгоритм интегрирования напряжений. Рассматривается модель изотропного материала без учёта скоростей деформирования с критерием текучести Мизеса для случая ПНС [29].

Критерий текучести Мизеса для ПНС²⁴:

$$\sigma_{11}^2 + \sigma_{22}^2 - \sigma_{11}\sigma_{22} + 3\sigma_{23}^2 = \sigma_y^2, \quad (19)$$

где σ_y – предел текучести, полученный из экспериментов на одноосное растяжение.

Для УПМ полная деформация складывается из упругой и пластической деформации. Рассмотрим модель Ramberg-Osgood:

$$\varepsilon = \varepsilon_e + \varepsilon_p = \frac{\sigma}{E} + \left(\frac{\sigma}{K}\right)^{1/n}, \quad (20)$$

или, пренебрегая первым слагаемым справа, получаем степенной закон течения (Power-Law):

$$\sigma_p(\varepsilon_p) = K(\varepsilon_p + \varepsilon_e)^n, \quad (21)$$

где E – модуль Юнга, K (коэффициент упрочнения или модуль пластического упрочнения) и n (показатель упрочнения) – константы, найденные в экспериментах на растяжение; ε_p – текущая пластическая деформация; ε_e – предельная упругая деформация:

$$\varepsilon_e = \left(E/K\right)^{1/(n-1)}. \quad (22)$$

Абедраббо и др. реализовали вычислительную схему (рис. 19), опираясь на теорию течения (incremental theory of plasticity). Процедура интегрирования напряжений основана на методе разделения переменных. Предварительные упругие напряжения (trial stress state) вычисляются выражением (18). Затем, проверяется условие текучести (23) и определяется тип состояния: упругость или пластика:

$$\phi = \sigma_{eq} - \sigma_p \leq 0, \quad (23)$$

где σ_{eq} – эквивалентные напряжения (например, по Мизесу); σ_p – напряжение упрочнения (hardening stress) (21).

Невыполнение условия (23) означает, что материал находится в пластической зоне, тогда запускается пластический корректор, который использует итеративный метод Ньютона для возврата предварительных упругих напряжений внутрь поверхности текучести, см. рис. 19.

В схеме численной процедуры итеративного интегрирования упругопластического материала с ассоциированным законом течения используются соотношения табл. 10.

24 http://en.wikipedia.org/wiki/Von_Mises_yield_criterion

Таблица 10. Соотношения, используемые для реализации УПМ (подробнее [30])

$\Phi = \sigma_{11}^2 + \sigma_{22}^2 - \sigma_{11} \sigma_{22} + 3 \sigma_{23}^2 - \sigma_y^2,$	Функция текучести (поверхность текучести, yield surface)	
$\sigma_p(\epsilon_p) = K(\epsilon_p) = K(\epsilon_p + \epsilon_e)^n,$	Закон упрочнения (hardening rule)	
$\epsilon_e = (E/K)^{[1/(n-1)]},$	Предельная упругая деформация	
$\dot{\sigma} = C : \dot{\epsilon}_{n+1}^e,$	Матрица упругих констант материала (material matrix)	
$\dot{\epsilon}^p = \dot{\lambda} \frac{\partial \Phi}{\partial \sigma},$	Ассоциативный закон течения (plastic multiplier)	
$\phi \leq 0,$	Критерий текучести (yield function)	
$\dot{\epsilon}_{n+1}^e = \dot{\epsilon}_{n+1},$	Приращение полной деформации берём за предварительное приращение упругой	
$\dot{\lambda} \geq 0,$	Normality parameter	Условия Куна-Таккера ²⁵
$\dot{\lambda} \Phi = 0,$	Khun-Tucker condition	
$\dot{\lambda} \dot{\Phi} = 0,$	Consistency condition	

где σ , $\dot{\epsilon}^e$ и $\dot{\epsilon}^p$ есть напряжение, скорости упругой и пластической деформация соответственно; C – тензор упругости 4-го порядка, константа; $\dot{\lambda}$ – коэффициент пластичности в ассоциативном законе течения. Проверка критерия текучести и условия нагружения-разгрузки осуществляются через условия Куна-Таккера.

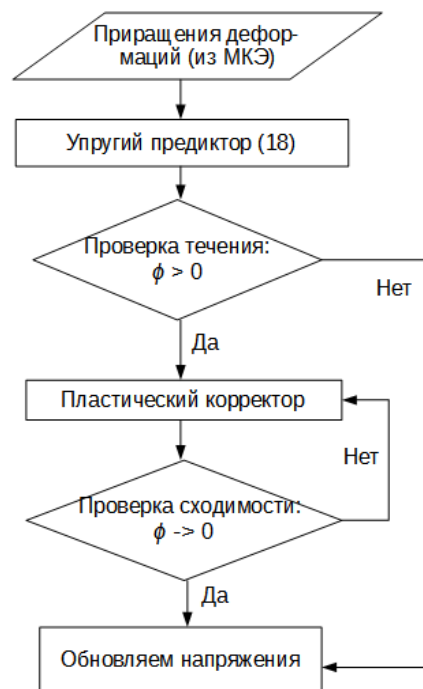


Рис. 19. Блок-схема процесса вычисления напряжений для УПМ

Метод секущих плоскостей в реализации Абедраббо применим в общем случае к любой функции течения. Функция течения по Мизесу м.б. явно найдена аналитически, что не требует использовать return mapping algorithm для размещения напряжений на поверхности текучести.²⁶

25 См. теорию методов оптимизации.

26 При решении нелинейной проблемы в МКЭ напряжения и деформации в точках интегрирования обычно получают интегрированием нелинейного уравнения движения различными алгоритмами и процедурами. Simo и Ortiz на основе полной деформационной теории реализовали схему интегрирования напряжений в УПМ при помощи метода секущих плоскостей, но алгоритм оказался медленным, см. в [21, 29] список источников.

5.3. Полимер для термомеханических задач

См. код примера Полимер для термомеханических задач [22]. Код работает только с твёрдотельными элементами.

Скорости нормальных деформаций:

$$\dot{\epsilon}_{ii} = \frac{\Delta \epsilon_{ii}}{\Delta t} \quad (24)$$

Скорости сдвиговых деформаций:

$$\dot{\epsilon}_{ij} = \frac{1}{2} \dot{\gamma}_{ij} = \frac{\Delta \gamma_{ij}}{2 \Delta t} \quad (25)$$

Тензор вязкости зависит от скорости сдвига и температуры (опционально). Зависимость скорости сдвига от вязкости представлена функцией Yasuda, также известная как обобщённая функция Cross/Carreau

$$\mu_{ij}(\dot{\gamma}_{ij}) = \mu_{\infty} + (\mu_0 - \mu_{\infty}) (1 + (k \dot{\gamma}_{ij})^a)^{(n-1)/a} \quad (26)$$

где μ_{∞} – нижняя граница вязкости при бесконечно большой скорости сдвига,

μ_0 – верхняя граница вязкости при нулевой скорости сдвига,

k – константа скорости реакции (characteristic time),

a, n – константы материала.

Если добавляется зависимость от температуры, то вязкость находится из уравнения Аррениуса

$$\mu_0(T) = A \exp\left(\frac{E_a}{RT}\right), \quad (27)$$

где A – константа материала, E_a – энергия активации, R – универсальная газовая постоянная, T – температура.

Объёмная вязкость зависит от усреднённой нормальной вязкости в соответствии с гипотезой Стокса:

$$\mu_B = -\frac{2}{3} \frac{\mu_{kk}}{3}. \quad (28)$$

Объёмной вязкостью часто можно пренебречь, а значение соответствующей константы в ключевом слове УМ обнулить.

Алгоритм в реализации Абедраббо, являясь одним из методов секущих плоскостей, которые в теории оптимизации принадлежат классу симплекс-алгоритмов, был разработан для того, чтобы избавиться от необходимости находить градиенты функции текучести и закона течения, как того требует the closest point projection iterative methods. Общая процедура the closest point projection обычно приводит к системам нелинейных уравнений, решение которых алгоритмом Ньютона-Рафсона требует вычисления градиентов системы уравнений. Хотя такой подход хорош для несложных моделей пластичности (например, Мизеса), его применение к такой сложной функции текучести, как Barlat96, чрезвычайно невыгодно для промышленного применения из-за увеличения количества итераций и общей медлительности.

Реализация Абедраббо базируется на следующем. В формулировке МКЭ по перемещениям МКЭ-код управляем по деформациям. Метод секущих плоскостей основан на методе разделения переменных, когда упругая и пластическая деформация разделяются. Метод Simo и Ortiz основан на полной деформационной теории, где значения полной деформации и полной пластической деформации на предыдущих шагах сохраняются для использования на следующем шаге, что не всегда обязательно, но для случая разгрузки улучшает точность решения. Использование the incremental theory of plasticity исключает данный шаг. В таких коммерческих кодах, как LS-Dyna и Abaqus, приращение деформации $\dot{\epsilon}_{n+1}$, предыдущее значение напряжений σ_n и любая переменная истории, сохранённые на предыдущем шаге, предназначены для обновления напряжений в начале текущего шага. Новое приращение деформации предполагается упругим и предиктор упругих напряжений “trial stress” работает с упругим законом. При использовании метода секущих плоскостей, текущее напряжённое состояние затем восстанавливается (пластический корректор) и вычисляются переменные пластичности.

Тензор вязких напряжений:

$$\tau_{ij} = 2\mu_{ij}\dot{\epsilon}_{ij} - \mu_B\dot{\epsilon}_{kk}\delta_{ij}. \quad (29)$$

Давление высчитывается из гидростатических напряжений на предыдущем шаге по времени и из приращения объёмных деформаций на текущем шаге:

$$p = \frac{-\sigma_{kk}}{3} - B\epsilon_{kk}. \quad (30)$$

Полный тензор напряжений:

$$\sigma_{ij} = \tau_{ij} - p\delta_{ij}, \quad (31)$$

где $B = E/(3(1-2\nu))$ – объёмный модуль, E – модуль Юнга, ν – коэффициент Пуассона. Напомним, что модуль сдвига $G = E/(2(1+\nu))$.

В ключевом слове `*MAT_USER_DEFINED_MATERIAL_MODELS` задаются следующие константы материала: `RO` – плотность, `YOUNG` – модуль Юнга, `POISSON` – коэффициент Пуассона, `BULKMOD` – объёмный модуль, `SHEARMOD` – модуль сдвига, `BULKVISC` – объёмная вязкость, `SHTIME` – константа k в выражении (26), `N-EXP` – n (26), `A-EXP` – a (26), `MUINF` – μ_∞ (26), `MU0` – μ_0 (26), `A` – A (27), `ACT-E` – E_a (27), `MUMAX` – верхний численный предел для вязкости, `BULKVMIN` – нижний численный предел для объёмной вязкости.

Пример: подключение полистирола, изотермический процесс при 453 К, вязкость высчитывается с помощью выражения (26), `SEI` – мм и МПа (плотность увеличена для сокращения времени счёта):

```
*MAT_USER_DEFINED_MATERIAL_MODELS
$ MID RO MT LMC NHV IORTHO IBULK IG
1,1E-05,43,16,50,0,3,4
$ IVECT IFAIL ITHERM IHYPER IEOS
1,0,1,1
$ YOUNG POISSON BULKMOD SHEARMOD
$ [MPa] [-] [MPa] [MPa]
2.4e3,0.3,2.0e3,9.23e2
$ SHTIME N-EXP A-EXP MUINF MU0 MUMAX BULKVMIN
$ [s] [-] [-] [MPa s] [MPa s] [MPa s] [MPa s]
5.18E+000,4.71E-001,2,7.00E-005,7.82E-002,7.82E-002,0
```

Пример: подключения полистирола в диапазоне рабочих температур от 453 К до 531 К, вязкость высчитывается с помощью выражения (27), `SEI` – м и Па:

```
*MAT_USER_DEFINED_MATERIAL_MODELS
$ MID RO MT LMC NHV IORTHO IBULK IG
1,1E+05,43,16,50,0,3,4
$ IVECT IFAIL ITHERM IHYPER IEOS
1,0,1,1
$ YOUNG POISSON BULKMOD SHEARMOD
$ [Pa] [-] [Pa] [Pa]
2.4e9,0.3,2.0e9,9.23e8
$ SHTIME N-EXP A-EXP MUINF A ACT-E MUMAX BULKVMIN
$ [s] [-] [-] [Pa s] [Pa s] [J/mol] [Pa s] [Pa s]
5.968778,.4815149,2,70,7.573e-11,1.305e+05,7.82e+04,-5.21e+04
```

6. Верификация УМ

Верификация (проверка работоспособности) подпрограммы УМ шаг важный и необходимый. Проверка подразумевает решение простейших задач с новым материалом и сравнение полученных результатов с известными аналитическими или численными решениями, как вариант, сравнением с поведением стандартных материалов в той же LS-DYNA. Постановки модельной задачи могут отличаться в зависимости от типа материала (изотропный-анизотропный), типа теста (сжатие-растяжение...). Например, анизотропные модели Хилла и Барлата могут сводиться к изотропным при единичном параметре пластической анизотропии R , тогда для растягиваемого стандартного образца-пластины отношение истинных деформаций по ширине модели к истинным деформациям по толщине модели должно быть равно единице, в то время, как для анизотропного материала отношение будет иным [29]).

Приведём верификационные модели Надера Абедраббо [29]. Ключевые файлы рассмотренных далее примеров (singleshell.k, tensile.k) доступны для скачивания на сайте Н. Абедраббо, впрочем, несложно составить их самостоятельно.

6.1. Верификация линейно-упругого материала

Используется материал Примера 1. Имеем единственный элемент оболочки размером 1x1 мм, см. рис. 20. Один из узлов элемента зафиксирован, в двух узлах ограничено по одной степени свободы для осуществления скольжения узлов вдоль Ox и Oy .

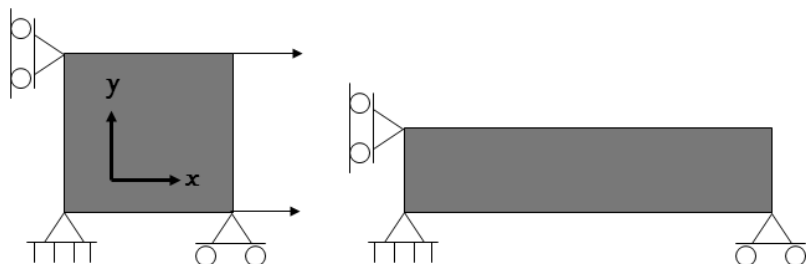


Рис. 20. Элемент оболочки в недеформированном и деформированном состоянии. Источник: [29]

Получим аналитическое решение для рассматриваемой задачи. Требуется наложить дополнительные граничные условия на модель так, чтобы итоговая истинная деформация ϵ_x приняла значение равное единице. В численном решении ожидаем получить такой же результат.

Истинная деформация ϵ_x и инженерная деформация e связаны соотношением:

$$\epsilon_x = \ln(1+e) = 1, \quad (32)$$

откуда инженерная деформация

$$e = \exp(1) - 1 = 2.71828 - 1 = 1.71828. \quad (33)$$

Как известно, инженерные деформации определяются через начальную l_0 и итоговую длину l . Найдём итоговую длину элемента вдоль Ox по найденной инженерной деформации.

$$e = \frac{l-l_0}{l_0} = \frac{l}{l_0} - 1; \quad 1.71828 = l/l_0 - 1; \quad (34)$$

$$\Rightarrow l = 1.71828 + 1 = 2.71828 \text{ мм.}$$

Таким образом, потребуется наложить на часть узлов модели ГУ по перемещениям $u_x = 1.71828$ мм. Решая задачу с УМ из Примера 1 получим, что итоговая деформация принимает значение 1, что удовлетворяет нашему аналитическому решению.

6.2. Верификация упруго-пластического материала с упрочнением

В данном примере мы удостоверимся в том, что метод интегрирования напряжений (Return Mapping Algorithm) корректно и точно накладывает текущее напряженное состояние на поверхность текучести в соответствии с функцией течения, заданной степенным законом (Power Law):

$$\sigma_p(\varepsilon_p) = K \cdot (\varepsilon_0 + \varepsilon_p)^n \quad (35)$$

Вспользуемся геометрией стандартного образца (размеры и граничные условия даны на рис. 21) и проведём тест на растяжение. Свойства материала: плотность $2.71E-6$ кг/мм³, модуль Юнга 69 ГПа, коэффициент Пуассона 0.34, константа $K=380$ МПа, показатель степени $n=0.21$, упругие деформации в момент начала пластического течения $\varepsilon_0=0.001382$ найдены по формуле (22).

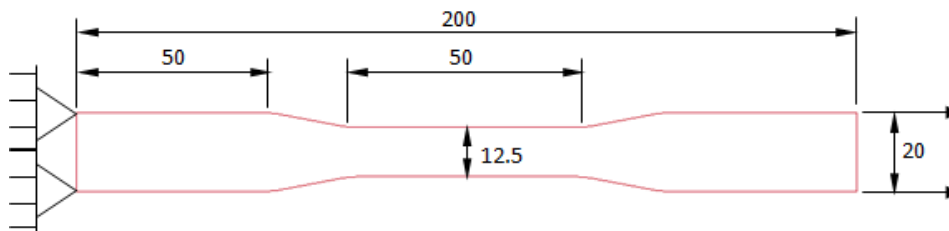


Рис. 21. Прямоугольный образец для теста на растяжение, в соответствии со стандартом ASTM-E8, размеры в мм. Источник: [29]

Подсчитаем теоретические значения истинных напряжений, используя заданные константы для степенного закона упрочнения (35) с шагом пластической деформации 0.01 в диапазоне от 0.0 до 0.3, и нанесём точки на график. Проведём численный эксперимент, получим кривую растяжения образца, выделив истинные напряжения и пластические деформации из НДС элемента в центре масс образца, и внесём численное решение на график, см. рис. 22.

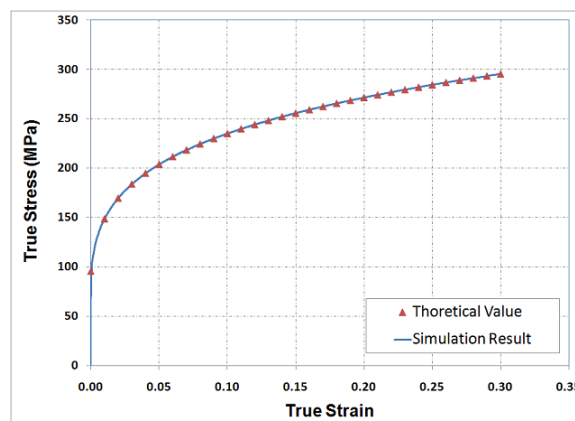


Рис. 22. Экспериментальная кривая растяжения и теоретические значения в истинных величинах. Источник: [29]

Как видно из графика, численные значения напряжений (сплошная линия) совпадают с теоретическими значениями (точки), что указывает на правильно составленный в UMAT алгоритм интегрирования напряжений и успешное возвращение напряжений на поверхность текучести.

Литература²⁷, ссылки на примеры и видеоролики²⁸ по основам LS-DYNA

1. Чернявский А.О. Метод конечных элементов. Основы практического применения.	A,L
2. Hallquist J.O. LS-DYNA 9.60 Manuals (in Russian)	L
3. Hallquist J.O. LS-DYNA Theory manual, 2006	L
4. Hallquist J.O. LS-DYNA User's Manuals. Livermore Software Technology Corporation, Livermore, 1999-2017.	L
5. LS-PREPOST User's Manual 4.2-4.3, LS-PREPOST Tutorial 4.2-4.3	L
6. Илюшкин М.В. Моделирование процессов обработки металлов давлением в программе ANSYS/LS-DYNA (осадка цилиндрической заготовки). Учебно-методическое пособие. – Ульяновск: УлГУ, 2013 – 112 с.	A,L
7. Маслов В.Д., Николаенко К.А. Моделирование процессов листовой штамповки в программном комплексе ANSYS/LS-DYNA.	A,L
8. Кравчук А.С. Кравчук А.В. Лекции по ANSYS/LS-DYNA и основам LS-PREPOST с примерами решения задач. В 3 ч. – Минск, БГУ, 2013.	A,L
9. Шляпугин А.Г. Моделирование и оптимизация технологических процессов ОМД. – Самарский государственный аэрокосмический университет, 2010. – 46 с.	A
10. ANSYS LS-DYNA Users Guides v11 Rus и v12.1 2009 Eng	A
11. http://www.dynasupport.com/tutorial	L
12. http://www.dynaexamples.com/	L
13. http://www.dynasupport.com/faq	L
14. http://www.lsdyna-tutorials.com/tutorials/basic	L
15. https://www.youtube.com/user/fzbn12/videos	L
16. https://www.youtube.com/channel/UCTwpjqCxKQMUckqXH8rEeFQ/videos	L
17. https://www.youtube.com/channel/UCMvJX7VH8KkeJwSmVbBlkmw/videos	L
18. https://www.youtube.com/channel/UCTwpjqCxKQMUckqXH8rEeFQ/videos	L
19. http://www.feaiej.com/ , http://www.feapublications.com/	L

Источники по UM и FLD

20. LS-DYNA custom constitutive subroutine basic process by df_af_aq [Электронный ресурс] / URL: forum.simwe.com, 2011-7-22.
21. Anucha Promwungkwa, Paiboon Limpitipanich Description of User-Defined Material Subroutine (UMAT) in LS-DYNA for Nonlinear Solid Mechanics [Электронный ресурс] / The 5th TSME International Conference on Mechanical Engineering 17-19th December 2014, The Empress, Chiang Mai
22. Vingaard M., Endelt B. Ø., Christiansen J.D.C. Implementation of a Material Model with Shear Rate and Temperature Dependent Viscosity [Электронный ресурс] / In Proceedings of 6th European LS-DYNA Users' Conference, 2007. pp. 5.213-5.222. URL: [http://vbn.aau.dk/en/publications/implementation-of-a-material-model-with-shear-rate-and-temperature-dependent-viscosity\(1d059d80-4f1f-11dc-a457-000ea68e967b\).html](http://vbn.aau.dk/en/publications/implementation-of-a-material-model-with-shear-rate-and-temperature-dependent-viscosity(1d059d80-4f1f-11dc-a457-000ea68e967b).html)

27 A,L – материалы по ANSYS/LS-DYNA, L – материалы по LS-DYNA.

28 Видеоролики являются собственностью их авторов, а именно: Dr. Ameen Tora, Ahmed Ali El-Hattab, кафедра «ТЗШП» при АО «УлНИАТ» и др.

23. Kalsi G.S. On the Experiences of Adding a Complex User-Material-Model to LS-DYNA [Электронный ресурс] / 12th International LS-DYNA Users Conference Constitutive Modeling(2). URL: <http://www.dynalook.com/international-conf-2012/constitutivemodeling12-b.pdf>
24. Rahmanian R. Optimizing mechanical stiffness of shape memory and superelastic alloys by introducing engineering porosity [Электронный ресурс] / Theses and Dissertations 1807, 2014. URL: http://utdr.utoledo.edu/theses-dissertations/1807/?utm_source=utdr.utoledo.edu%2Ftheses-dissertations%2F1807&utm_medium=PDF&utm_campaign=PDFCoverPages
25. Erhart T. An Overview of User Defined Interfaces in LS-DYNA [Электронный ресурс] / LS-DYNA May 23rd-24th, 2011. 8th European Users Conference STRASBOURG – France Session. URL: <https://www.dynamore.de/de/download/papers/konferenz11/papers/session22-paper1.pdf>
26. Borrvall T. A User-Defined Element Interface in LS-DYNA v971 [Электронный ресурс] / Proc. 9th International LS-DYNA Users Conference, Dearborn, MI, USA, 2006, 18-11ff. URL: <https://www.dynamore.de/de/download/papers/forum06/new-methods/a-user-defined-element-interface-in-ls-dyna-v971>
27. Rajesh K. User material Development in LS Dyna [Электронный ресурс] / Impact Lab, DME IIT Delhi. URL: https://www.slideshare.net/sanjerajesh/user-material-dev?qid=715b00f7-17fc-46a7-9a9b-f7c6bf6134ec&v=&b=&from_search=1
28. Новожилов Ю. User Defined Materials for LS-DYNA [Электронный ресурс] / URL: <https://www.slideshare.net/YuryNovozhilov/user-defined-materials-in-lsdyna>
29. Abedrabbo N. The FORTRAN implementation of the von Mises User Material Subroutine along with the stress integration procedure. Theoretical development and verifications [Электронный ресурс] / URL: <https://sites.google.com/site/aenader/umat-workshop>
30. Abedrabbo N.E. Forming of aluminum alloys at elevated temperatures. A dissertation submitted to Michigan State University in partial fulfillment of the requirements for the degree of Doctor Philosophy / Department of Mechanical Engineering, 2005.
31. LS-DYNA 常见问题汇总 2.0 [LS- DYNA Frequently Asked Questions Summary 2.0] by yuminhust2005 [Электронный ресурс] / SimWe Simulation Forum <forum.simwe.com>
32. Zhao H. LS-DYNA Dynamic Analysis Guide [M]. Beijing: Press of Weapon Industry, 2003. (in Chinese)
33. Han Z., Wainscott B. Introduction to the new Framework for User Subroutine Development of LS-DYNA [Электронный ресурс] / FEA Information Engineering Journal, Volume 6, Issue Q1, February 2017. URL: http://www.feaiej.com/2017/feaiej_q1_2017.pdf
34. ASM HandBook. Volume 14 – Forming and Forging. 9th Edition Metals Handbook. - ASM International, 2110 p.
35. Качанов Л.М. Основы механики разрушения. - Главная редакция физико-математической литературы изд-ва «Наука», 1974. - стр. 312.
36. Jday's Frequently Asked Questions: Use Defined Materials [Электронный ресурс] / URL: http://ftp.lstc.com/anonymous/outgoing/jday/faq/user_defined_materials.faq
37. Panich S., Uthaisangsuk V., Juntarat J. and Suranuntchai S. Determination of Forming Limit Stress Diagram for Formability Prediction of SPCE 270 Steel Sheet / Journal of Metals, Materials and Minerals, Vol.21 No.1 pp.19-27, 2011.
38. Арсентьева К.С., Кункин С.Н., Мамутов А.В., Мамутов В.С. Диаграммы предельных деформаций тонколистового металла при формовке подвижными средами / Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2(219)' 2015, С. 159-166.
39. Balod A.O. Theoretical Determination of Forming Limit Stress Diagram for Steel 1021 Sheet | Tikrit Journal of Eng. Sciences / Vol.17 / No.4 / December 2010, (56-66).
40. Graf A., Hosford W. F. Calculation of Forming Limit Diagrams / Metallurgical transactions A / Vol. 21A / January 1990, (87-94).
41. Şanay B. Prediction of Plastic Instability and Forming Limits in Sheet Metal Forming / A Thesis Submitted to the Graduate School of Natural and Applied Sciences of Middle East Technical University in Partial Fullfilment of the Requirements for the Degree of Master of Science in Mechanical Engineering / September 2010, 149 p.

42. Bruschi S., Altan T., Banabic D. et al. Testing and modelling of material behaviour and formability in sheet metal forming / CIRP Annals - Manufacturing Technology 63 (2014) 727–749
43. Изосимова С.В. Исследование влияния формы заготовки на точность построения диаграммы предельных деформаций [Электронный ресурс] / Всероссийская научно-техническая конференция студентов Студенческая научная весна 2013: Машиностроительные технологии. URL: http://studvesna.ru/db_files/articles/810/article.pdf
44. Akhtar S. Khan, Muneer Baig Anisotropic responses, constitutive modeling and the effects of strain-rate and temperature on the formability of an aluminum alloy / International Journal of Plasticity 27 (2011) 522–538.
45. Davies R.W., Grant G.J., Oliver H.E., Khaleel M.A., Smith M.T. Forming-Limit Diagrams of Aluminum Tailor-Welded Blank Weld Material / Metallurgical And Materials Transactions A. Volume 32a, February 2001, 275-283.
46. Silva M.B., Isik K., Tekkaya A.E., Martins P.A.F. Fracture Loci in Sheet Metal Forming: A Review / Acta Metall. Sin. (Engl. Lett.), 2015, 28(12), 1415–1425.
47. Kotkunde N., Srinivasan S., Krishna G., Gupta A.K., Singh S.K. Influence of material models on theoretical forming limit diagram prediction for Ti-6Al-4V alloy under warm condition / Trans. Nonferrous Met. Soc. China 26(2016) 736–746.
48. Nishant J. Introduction to LS-DYNA / CROWN Technology, 2006.
49. Cordasco D., Emmerling W., Bois P.D. A Status Review of Failure Simulation at the Federal Aviation Administration.
50. Ozarmut B., Richter H., Brosius A. Probabilistic Analysis of Process Chain “Forming to Crash” Regarding Failure Prediction.
51. Fleischer M., Lipp A., Meinhardt J., Hippchen P., Heinle I., Ickes A., Senner T. Usage of LS-DYNA in metal forming / 10th European LS-DYNA Conference – 15. - 17.6.2015.
52. Haight S., Kan C.D., Bois D.P. Development of a Fully-Tabulated, Anisotropic and Asymmetric Material Model for LS-DYNA (*MAT_264) / 10th European LS-DYNA Conference 2015, Würzburg, Germany, 2015.
53. Dutton T. Review of Sheet Metal Forming Simulation Progress to Date, Future Developments / Dutton Simulation Ltd. 8th European LS-DYNA Users Conference, Metal Forming.
54. Onipede J.O., Gomes C.J. Springback in High Strength Anisotropic Steel
55. Radonjic R., Liewald M., Han F. Optimisation of the blank holder stiffness in deep drawing processes by using FEA / 10th European LS-DYNA Conference 2015, Würzburg, Germany, 2015.
56. LS-PREPOST Online Documentation. Tutorial 17: Introduction to MetalForming Interface [Электронный ресурс] / Forming Version 1.0 April 2009. / URL: http://www.lstc.com/lssp/content/tutorials/17/MFGUI_Tutorial_Forming.pdf, http://www.lstc.com/lssp/content/tutorials/17/MFGUI_Tutorial_Forming_Files.zip
57. Neukamm F., Feucht M., Haufe A. Considering damage history in crachworthiness simulations / 7th European LS-DYNA Conference.
58. Kumar S.D., Jeyasingh J.J.V., Amjith T.R Development of Nakazima Test Simulation Tool for Forming Limit Diagram Generation of Aluminium Alloys / International Journal of Engineering Studies and Technical Approach, Volume 01, No. 10, October 2015.
59. Maier C. Failure prediction in sheet metal forming using fea of Nakazima test / Machines, technologies, materials. ISSN 1313-0226. ISSUE 7/2013.
60. Кайбаров А. П. Исследование влияния формы перетяжного ребра на требуемую силу прижима и разработка конструкции штампа для построения кривой предельных деформаций [Электронный ресурс] / Всероссийская научно-техническая конференция студентов Студенческая научная весна 2016: Машиностроительные технологии. URL: <http://studvesna.ru>
61. Panich S., Uthaisangsuk V. Influences of yield criteria on stress based forming limit diagram of DP980 steel / XIII International Conference on Computational Plasticity. Fundamentals and Applications COMPLAS XIII E. Oñate, D.R.J. Owen, D. Peric and M. Chiumenti (Eds).
62. Khalatbari H. Investigation of Formability of Material in Incremental Sheet Metal Forming Process. / Submitted to the Institute of Graduate Studies and Research in partial fulfillment of the requirements for the Degree of Master of Science in Mechanical Engineering / Eastern Mediterranean University. Gazimağusa, North Cyprus. September 2012.

63. Behrens B.-A., Guidault P.-A., Homann B. Development of a friction approach for the FE method of sheet metal forming based on multi-scale modeling / International Research Training Group „Virtual Materials and Structures and their Validation” – IRTG 1627.
64. Ferrero S., Tribuzi C., DiNatale F. Sheet metal forming optimization using ANSA and LS-DYNA / 6th BETA CAE International Conference.
65. Neukamm F., Feucht M., Haufe A., Roll K. On closing the Constitutive Gap Between Forming and Crash Simulation / 10th International LS-DYNA Users Conference.
66. Bois P.D, Buyuk V., He J., Kan S. Development, implementation and Validation of 3-D Failure Model for Aluminium 2024 for High Speed Impact Applications / 11th International LS-DYNA Users Conference June 6-8 2010 Dearborn, Michigan, USA.
67. Memhard D., Andrieux F., Sun D.-Z., Häcker R. Development and verification of a material model for prediction of containment safety of exhaust turbochargers / 8th European LS-DYNA Users Conference, Strasbourg - May 2011.
68. Current research activities at HAMK Sheet Metal Development Centre / MetNet Workshop Hämeenlinna, 2nd February 2011. 6 p.
69. ISO 12004-2:2008 Metallic materials - Sheet and strip - Determination of forming-limit curves - Part 2: Determination of forming-limit curves in the laboratory
70. Atli-Veltin B., Vredevelde L. Use of Forming Limit Curve as a Failure Criterion in Maritime Crash Analysis / 10th European LS-DYNA Conference 2015, Würzburg, Germany.
71. Bi J. Constitutive modeling of aluminum foam and finite element implementation for crash simulations. A dissertation submitted to the faculty of The University of North Carolina at Charlotte in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering. Charlotte 2012. 130 p.
72. Varmint AI's material database. URL: <http://www.varmintal.com/aengr.htm>

Литература по вычислительной механике

73. Образцов И.Ф., Савельев Л.М., Хазанов Х.С. Метод конечных элементов в задачах строительной механики летательных аппаратов: Учеб. пособие для студентов авиац. спец. вузов. - М.: Высш. шк., 1985. - 392 с., ил.
74. Kwon, Young W. The finite element method using MATLAB, 4 ed. / by Young W. Kwon and Hyochoong Bang. - CRC Press LLC, 1997. - 527 p.
75. Simo J.C., Hughes T.J.R. Computational Inelasticity. Series: Interdisciplinary Applied Mathematics, Vol. 7, 1998., 392 pages.
76. Crisfield M.A. Non-Linear Finite Element Analysis of Solids and Structures, Volume 1, Essentials, July 1998.
77. Crisfield M.A. Non-Linear Finite Element Analysis of Solids and Structures, Volume 2, Essentials, July 1997, 494 p.
78. Borst R., Crisfield M.A., Remmers J.J.C., Verhoosel C.V. Non-linear Finite Element Analysis of Solids and Structures / John Wiley & Sons Ltd, 2012. 516 p.
79. Kobayashi S., Oh S., Altan T. Metal Forming and the Finite-Element Method \ Oxford University Press, 1989. 377 p.
80. Гун Г.Я. Математическое моделирование процессов обработки металлов давлением. Учебное пособие для вузов. М., «Металлургия», 1983. 352 с.
81. Гун Г.Я. Теоретические основы обработки металлов давлением. (Теория пластичности), Учебник для вузов. М., «Металлургия», 1980. 456 с.
82. Тарновский И.Я., Поздеев А.А., Ганаго О.А., Колмогоров В.Л. и др. Теория обработки металлов давлением (Вариационные методы расчета усилий и деформации). - М.: Государственное научно-техническое издательство литературы по черной и цветной металлургии, 1963. – 673 с.
83. Хечумов Р.А., Кепплер Х., Прокопьев В.И. Применение метода конечных элементов к расчету конструкций: Учебное пособие для технических вузов – М.: Издательство Ассоциации строительных вузов, 1994. – 353 с., ил.

7. Исходные тексты UM на языке FORTRAN

Примеры 7.1-7.5 являются стандартными, они содержатся в Keyword Manual Appendix A и в архиве с исходниками любой версии LS-DYNA.

Источник примера 7.6 [29].

Источник примера 7.7 [22].

Автор примеров 7.8, 7.9, 7.10 - Al Tabiei, см. <http://www.lsdyna-online.com/>.

Источник кода 7.11 (Hill's normal anisotropic plasticity) не установлен, вероятно, это часть внутренних исходников LS-DYNA или часть бесплатного кода DYNA3D.

Дополнительно самостоятельно см. следующие реализации UM:

1) модель льда по Друкеру-Прагеру: Fangming Heng Drucker-Prager bilinear isotropic hardening model development. Experimental and numerical modeling of high speed ice impact onto rigid target / Master degree by research. School of aerospace, transport and manufacturing. Applied Mechanics. 03/2015. - 133 p.

2) модель мышцы: Kleinbach C., Martynenko O., Promies J., Haeufle D.F.B., Fehr J., Schmitt S. Implementation and Validation of the Extended Hill-type Muscle Model with Robust Routing Capabilities in LS-DYNA for Active Human Body Models, Biomedical Engineering Online, 2017.

3) бетон с разрушением: P. Grassl, D. Xenos, U. Nyström, R. Rempling, K. Gylltoft.: CDPM2: A damage-plasticity approach to modelling the failure of concrete. International Journal of Solids and Structures. Volume 50, Issue 24, pp. 3805-3816, 2013; P. Grassl and M. Jirasek Damage-plastic model for concrete failure. International Journal of Solids and Structures. Vol. 43, pp. 7166-7196, 2006.

7.1. Линейно-упругий материал

```

c      implicit real*8 (a-h,o-z)
c      ****
c|      Livermore Software Technology Corporation (LSTC)      |
c|      -----|
c|      Copyright 1987,1988,1989 John O.Hallquist, LSTC      |
c|      Copyright 1987-2008 Livermore Software Tech. Corp    |
c|      All rights reserved                                  |
c      ****
c
c      изотропный упругий материал (пример пользовательской подпрограммы)
c
c      Переменные:
c      cm(1)=первая константа материала, в данном случае модуль Юнга
c      cm(2)=вторая константа материала, в данном случае коэффициент Пуассона
c      ...
c      cm(n)=n-я константа материала
c
c      eps(1)= x  -компонента локальных деформаций (local deformation)
c                                     (strain increment)
c      eps(2)= y  -компонента локальных деформаций
c      eps(3)= z  -компонента локальных деформаций
c      eps(4)= xy -компонента локальных деформаций
c      eps(5)= yz -компонента локальных деформаций
c      eps(6)= zx -компонента локальных деформаций
c
c      sig(1)= x  -компонента полных напряжений (total stress)
c      sig(2)= y  -компонента полных напряжений
c      sig(3)= z  -компонента полных напряжений
c      sig(4)= xy -компонента полных напряжений
c      sig(5)= yz -компонента полных напряжений

```



```

с      sig(6)= zx -компонента полных напряжений
с
с      hisv(1)=1-я переменная истории
с      hisv(2)=2-я переменная истории
с      ...
с      hisv(n)=n-я переменная истории
с
с      dt1=текущий шаг по времени
с      capa=коэффициент коррекции поперечного сдвига
с      etype:
с      eq."solid" ("brick" в ранних релизах) для твёрдотельных элементов
с      eq."sld2d" для оболочек с формулировками 13, 14, 15 (2D-твёрдотелы)
с      eq."shl_t" для оболочек с ф-ми 25, 26, 27 (shells with thickness stretch)
с      eq."shell" для прочих оболочечных элементов + плит с формулировками 1 и 2
с      eq."tshel" для элементов плит (tick shells), формулировки 3 и 5
с      eq."hbeam" для балочных элементов, формулировки 1 и 11
с      eq."tbeam" для ферменных балочных элементов, формулировка 3
с      eq."dbeam" для дискретных балочных элементов, формулировка 6
с      eq."beam" для всех прочих балочных элементов
с
с      tt=time=текущее время в задаче
с      temper=текущая температура
с
с      faillel=флаг разрушения, установить значение .true. чтобы разрушить элемент;
с      если уже установлен в .true. - элемент разрушен ранее
с
с      crv=массив из расчётного файла с некоторой табличной функцией
с
с      sma=массив с дополнительными константами материала
с
с      Переход в локальную систему элемента осуществляется перед вызовом
с      данной подпрограммы. Обратный переход в глобальную СК осуществляется после
с      выхода из данной подпрограммы.
с
с      Все переменные истории на первом шаге инициализируются в ноль.
с      Переменные истории могут быть установлены в ноль
с      во время первого вызова данной подпрограммы для каждого из элементов.
с
с      В dyna3d (LS-DYNA) вычисление энергии для подсчёта общего баланса энергии
с      осуществляются вне данной подпрограммы.
с
с      subroutine umat41(cm,eps,sig,eps,hsv,dt1,capa,etype,tt,temper,faillel,crv,sma)
с
с      include 'nlqparm'
с      include 'iounits.inc'
с      common/bk06/idmmy,iaddp,ifil,maxsiz,ncycle,time(2,30) ! include 'bk06.inc'
с      dimension cm(*),eps(*),sig(*),hsv(*),crv(101,2,*),sma(*)
с      character*5 etype
с      logical faillel
с
с      if (ncycle.eq.1) then
с          call usermsg('mat41')
с      endif
с*****
с      Начало кода UM
с*****
с      Обновление деформаций и напряжений
с      для твёрдотельников (brick/solid), оболочек и балок
с      Подсчёт модуля сдвига G
с          g2 = abs(cm(1))/(1.+cm(2))
с          g =.5*g2
с      ! если тип элемента единственный - оставьте блок кода в соответствующем if'e
с      ! if (etype.eq.'solid') then          !if (etype.eq.'brick') then
с          if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
с      *      etype.eq.'sld2d'.or.etype.eq.'tshel') then

```

```

    if (.not.failed) then
      davg=(-eps(1)-eps(2)-eps(3))/3.
      p=-davg*abs(cm(1))/(1.-2.*cm(2))
      sig(1)=sig(1)+p+g2*(eps(1)+davg)
      sig(2)=sig(2)+p+g2*(eps(2)+davg)
      sig(3)=sig(3)+p+g2*(eps(3)+davg)
      sig(4)=sig(4)+g*eps(4)
      sig(5)=sig(5)+g*eps(5)
      sig(6)=sig(6)+g*eps(6)
      if (cm(1).lt.0.) then
        if (sig(1).gt.cm(5)) failed=.true.
      endif
    endif
  elseif (etype.eq.'shell') then
    if (.not.failed) then
      gc = capa*g
      q1 = cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
      q3 = 1./(q1+g2)
      eps(3) = -q1*(eps(1)+eps(2))*q3
      davg = (-eps(1)-eps(2)-eps(3))/3.
      p = -davg*abs(cm(1))/(1.-2.*cm(2))
      sig(1) = sig(1)+p+g2*(eps(1)+davg)
      sig(2) = sig(2)+p+g2*(eps(2)+davg)
      sig(3) = 0.0
      sig(4) = sig(4)+g*eps(4)
      sig(5) = sig(5)+gc*eps(5)
      sig(6) = sig(6)+gc*eps(6)
      if (cm(1).lt.0.) then
        if (sig(1).gt.cm(5)) failed=.true.
      endif
    endif
  elseif (etype.eq.'beam') then
    q1 = cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
    q3 = q1+2.0*g
    gc = capa*g
    deti = 1./(q3*q3-q1*q1)
    c22i = q3*deti
    c23i = -q1*deti
    fac = (c22i+c23i)*q1
    eps(2)=-eps(1)*fac-sig(2)*c22i-sig(3)*c23i
    eps(3)=-eps(1)*fac-sig(2)*c23i-sig(3)*c22i
    davg = (-eps(1)-eps(2)-eps(3))/3.
    p = -davg*cm(1)/((1.-2.*cm(2)))
    sig(1)=sig(1)+p+g2*(eps(1)+davg)
    sig(2)=0.0
    sig(3)=0.0
    sig(4)=sig(4)+gc*eps(4)
    sig(5)=0.0
    sig(6)=sig(6)+gc*eps(6)
  elseif (etype.eq.'tbeam') then
    q1 = cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
    q3 = q1+2.0*g
    deti = 1./(q3*q3-q1*q1)
    c22i = q3*deti
    c23i = -q1*deti
    fac = (c22i+c23i)*q1
    eps(2)=-eps(1)*fac
    eps(3)=-eps(1)*fac
    davg = (-eps(1)-eps(2)-eps(3))/3.
    p = -davg*cm(1)/(1.-2.*cm(2))
    sig(1)=sig(1)+p+g2*(eps(1)+davg)
    sig(2)=0.0
    sig(3)=0.0
  else
    write (*,10) etype

```

```

      write (iohsp,10) etype
      write (iomsg,10) etype
!     call adios(2)
!!    cerdat(1)=etype
!!    call lsmsg(3,MSG_SOL+1150,ioall,ierdat,rerdat,cerdat,0)
      end if
c
c 10    format(/
        1 ' *** Error element type ',a,' can not be',
        2 '                run with the current material model.')
```

c*****

c Конец кода UM

c*****

```

      return
      end
```

7.2. Линейный ортотропный материал (векторная подпрограмма)

```

c      subroutine umat42 (cm,eps,sig,eps,hsv,dtl,capa,etype,tt,
c      1 temper,failel,crv,cma,temper)
c      dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*),cma(*)
c      character*5 etype
c      logical failel
c      return
c      end
!
```

```

      subroutine umat42v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
* sig3,sig4,sig5,sig6,epsps,hsvs,lft,llt,dtlsiz,capa,
* etype,tt,temps,failels,nlqa,crv,cma)
!      subroutine umat43v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
!      * sig3,sig4,sig5,sig6,epsps,hsv,lft,llt,dtlsiz,capa,
!      * etype,tt,temps,failels,nlqa,crv,cma)
c
c*****
c|  Livermore Software Technology Corporation (LSTC)          |
c|  -----|
c|  Copyright 1987-2007 LSTC                                  |
c|  All Rights Reserved                                       |
c*****
      include 'nlqparm'
!      include 'bk06.inc'
!      include 'iounits.inc'
      dimension d1(*),d2(*),d3(*),d4(*),d5(*),d6(*)
      dimension sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*)
      dimension cm(*),epsps(*),hsvs(nlq,*),dtlsiz(*)
!      dimension cm(*),epsps(*),hsv(nlq,*),dtlsiz(*)
      dimension temps(*),crv(lq1,2,*),cma(*)
      logical failels(*)
      character*5 etype
c
c      eps(1)...eps(6)=приращения x,y,z,xу,yz,zx компонент локальных деформаций
c      sig(1)...sig(6)=локальные напряжения по x,y,z,xу,yz,zx
c
c Свойства упругого материала: если не заданы упругие свойства вдоль
c разных направлений, считается, что материал изотропный
c
      E1 = cm(1)
      E2 = cm(12)
      E3 = cm(13)
      v21 = cm(14)
      v31=cm(15)
      v32=cm(16)
      G12 = cm(17)
```

```

      G23 = cm(18)
      G31 = cm(19)
c
      v12 = v21*E1/E2
      v23 = v32*E2/E3
      v13 = v31*E1/E3
c Матрица упругих констант
      S = 1.-v12*v21-v23*v32-v31*v13-2.*v21*v32*v13
!      S = (1.-v12*v21-v23*v32-v31*v13-2*v21*v32*v13) / (E1*E2*E3) !второй вариант
c
      C11=(1. - v23*v32)*E1/S
      C12=(v21 + v31*v23)*E1/S
      C13=(v31 + v21*v32)*E1/S
c      C13=(v31 + v21*v23) / (E2*E3*S)
      C22=(1. - v31*v13)*E2/S
      C23=(v32 + v31*v12)*E2/S
      C33=(1. - v12*v21)*E3/S
!      C11=(1. - v23*v32) / (E2*E3*S) !второй вариант
!      C12=(v21 + v31*v23) / (E2*E3*S)
!      C13=(v31 + v21*v32) / (E2*E3*S)
!      C22=(1. - v31*v13) / (E1*E3*S)
!      C23=(v32 + v31*v12) / (E1*E3*S)
!      C33=(1. - v12*v21) / (E1*E2*S)
      C44=G12
      C55=G23
      C66=G31
c
c      write(59,*) 'Stiffness Matrix'
c      write(59,*) C11, ' ', C12, ' ', C13
c      write(59,*) C22, ' ', C23, ' ', C33
c      write(59,*) C44, ' ', C55, ' ', C66
c      write(59,*) ' '
c
      do 10 i=lft,llt
      if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
*      etype.eq.'sld2d'.or.etype.eq.'tshel') then
c Вычисление напряжений, вариант первый
      sig1(i) = sig1(i)+(C11*d1(i)+C12*d2(i)+C13*d3(i)) !*dt1siz(i) !второй вар.
      sig2(i) = sig2(i)+(C12*d1(i)+C22*d2(i)+C23*d3(i)) !*dt1siz(i)
      sig3(i) = sig3(i)+(C13*d1(i)+C23*d2(i)+C33*d3(i)) !*dt1siz(i)
      sig4(i) = sig4(i)+C44*d4(i) !*dt1siz(i)
      sig5(i) = sig5(i)+C55*d5(i) !*dt1siz(i)
      sig6(i) = sig6(i)+C66*d6(i) !*dt1siz(i)
c
      else if (etype.eq.'shell') then
c      hsv(i,7)=- (hsv(i,4)*C13*hsv(i,1) + C23*hsv(i,6))/C33
      d3(i) = -(C13*d1(i) + C23*d2(i))/C33
      sig1(i) = sig1(i)+(C11*d1(i)+C12*d2(i)+C13*d3(i))
      sig2(i) = sig2(i)+(C12*d1(i)+C22*d2(i)+C23*d3(i))
      sig3(i) = 0.0
      sig4(i) = sig4(i)+C44*d4(i)
      sig5(i) = sig5(i)+capa*C55*d5(i)
      sig6(i) = sig6(i)+capa*C66*d6(i)
      endif
      10 continue
c
! Второй вариант вычисления напряжений
      do 10 i=lft,llt
      if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
*      etype.eq.'sld2d'.or.etype.eq.'tshel') then
c
c ВЫЧИСЛИТЬ ШЕСТЬ КОМПОНЕНТ ТЕНЗОРА ДЕФОРМАЦИЙ
c      hsv(i,1)=hsv(i,1)+d1(i)
c      hsv(i,6)=hsv(i,6)+d2(i)
c      hsv(i,7)=hsv(i,7)+d3(i)

```

```

c      hsv(i,8)=hsv(i,8)+d4(i)
c      hsv(i,9)=hsv(i,9)+d5(i)
c      hsv(i,10)=hsv(i,10)+d6(i)
c      Вычислить напряжения
c      sig1(i) = C11*hsv(i,1)+C12*hsv(i,6)+C13*hsv(i,7)
c      sig2(i) = C12*hsv(i,1)+C22*hsv(i,6)+C23*hsv(i,7)
c      sig3(i) = C13*hsv(i,1)+C23*hsv(i,6)+C33*hsv(i,7)
c      sig4(i) = C44*hsv(i,8)
c      sig5(i) = C55*hsv(i,9)
c      sig6(i) = C66*hsv(i,10)
c
c      sig1(i) = sig1(i)+(C11*d1(i)+C12*d2(i)+C13*d3(i))
c      sig2(i) = sig2(i)+(C12*d1(i)+C22*d2(i)+C23*d3(i))
c      sig3(i) = sig3(i)+(C13*d1(i)+C23*d2(i)+C33*d3(i))
c      sig4(i) = sig4(i)+C44*d4(i)
c      sig5(i) = sig5(i)+C55*d5(i)
c      sig6(i) = sig6(i)+C66*d6(i)
c
c      S1 = C11*d1(i)+C12*d2(i)+C13*d3(i)
c      S2 = C12*d1(i)+C22*d2(i)+C23*d3(i)
c      S3 = C13*d1(i)+C23*d2(i)+C33*d3(i)
c      S4 = C44*d4(i)
c      S5 = C55*d5(i)
c      S6 = C66*d6(i)
c
c      hsv(i,1) = 2.0
c      hsv(i,2) = 0.9
c      hsv(i,3) = 1.2
c      hsv(i,4) = 0.5
c      hsv(i,5) = 3.0
c      hsv(i,6) = 5.0
c      hsv(i,7) = 0.7
c      hsv(i,8) = 3.4
c      hsv(i,9) = 8.0
c      Right Green-St. Venant Strain Tensor
c      G1 = 0.5*(hsv(i,1)*hsv(i,1)+hsv(i,2)*hsv(i,2)+hsv(i,3)*hsv(i,3)-1.)
c      G2 = 0.5*(hsv(i,4)*hsv(i,4)+hsv(i,5)*hsv(i,5)+hsv(i,6)*hsv(i,6)-1.)
c      G3 = 0.5*(hsv(i,7)*hsv(i,7)+hsv(i,8)*hsv(i,8)+hsv(i,9)*hsv(i,9)-1.)
c      G4 = 0.5*(hsv(i,1)*hsv(i,4)+hsv(i,2)*hsv(i,5)+hsv(i,3)*hsv(i,6))
c      G5 = 0.5*(hsv(i,4)*hsv(i,7)+hsv(i,5)*hsv(i,8)+hsv(i,6)*hsv(i,9))
c      G6 = 0.5*(hsv(i,1)*hsv(i,7)+hsv(i,2)*hsv(i,8)+hsv(i,3)*hsv(i,9))
c      Left Green-St. Venant Strain Tensor
c      G1 = 0.5*(hsv(i,1)*hsv(i,1)+hsv(i,4)*hsv(i,4)+hsv(i,7)*hsv(i,7)-1.)
c      G2 = 0.5*(hsv(i,2)*hsv(i,2)+hsv(i,5)*hsv(i,5)+hsv(i,8)*hsv(i,8)-1.)
c      G3 = 0.5*(hsv(i,3)*hsv(i,3)+hsv(i,6)*hsv(i,6)+hsv(i,9)*hsv(i,9)-1.)
c      G4 = 0.5*(hsv(i,1)*hsv(i,2)+hsv(i,4)*hsv(i,5)+hsv(i,7)*hsv(i,8))
c      G5 = 0.5*(hsv(i,2)*hsv(i,3)+hsv(i,5)*hsv(i,6)+hsv(i,8)*hsv(i,9))
c      G6 = 0.5*(hsv(i,1)*hsv(i,3)+hsv(i,4)*hsv(i,6)+hsv(i,7)*hsv(i,9))
c
c      write(59,*) 'Green-St. Venant Strain Tensor'
c      write(59,*) G1, ' ', G4, ' ', G6
c      write(59,*) G4, ' ', G2, ' ', G5
c      write(59,*) G6, ' ', G5, ' ', G3
c      write(59,*) ' '
c      Second Piola-Kirchhoff Stress Tensor
c      S1 = C11*G1 + C12*G2 + C13*G3
c      S2 = C12*G1 + C22*G2 + C23*G3
c      S3 = C13*G1 + C23*G2 + C33*G3
c      S4 = 2.*C44*G4
c      S5 = 2.*C55*G5
c      S6 = 2.*C66*G6
c
c      write(59,*) 'Second Piola-Kirchhoff Stress Tensor'
c      write(59,*) S1, ' ', S4, ' ', S6
c      write(59,*) S4, ' ', S2, ' ', S5

```

```

c      write(59,*) S6, ' ', S5, ' ', S3
c      write(59,*) ' '
c  Якобиан
      detF=hsv(i,1)*(hsv(i,5)*hsv(i,9)-hsv(i,6)*hsv(i,8))-
*      hsv(i,2)*(hsv(i,4)*hsv(i,9)-hsv(i,6)*hsv(i,7))+
*      hsv(i,3)*(hsv(i,4)*hsv(i,8)-hsv(i,5)*hsv(i,7))
c
c      write(59,*) 'Jacobian'
c      write(59,*) detF
c      write(59,*) ' '
c  Тензор напряжений Коши (F*S*F^T)
      FS11 = hsv(i,1)*S1 + hsv(i,4)*S4 + hsv(i,7)*S6
      FS12 = hsv(i,1)*S4 + hsv(i,4)*S2 + hsv(i,7)*S5
      FS13 = hsv(i,1)*S6 + hsv(i,4)*S5 + hsv(i,7)*S3
      FS21 = hsv(i,2)*S1 + hsv(i,5)*S4 + hsv(i,8)*S6
      FS22 = hsv(i,2)*S4 + hsv(i,5)*S2 + hsv(i,8)*S5
      FS23 = hsv(i,2)*S6 + hsv(i,5)*S5 + hsv(i,8)*S3
      FS31 = hsv(i,3)*S1 + hsv(i,6)*S4 + hsv(i,9)*S6
      FS32 = hsv(i,3)*S4 + hsv(i,6)*S2 + hsv(i,9)*S5
      FS33 = hsv(i,3)*S6 + hsv(i,6)*S5 + hsv(i,9)*S3
c
c      write(59,*) 'F*S'
c      write(59,*) FS11, ' ', FS12, ' ', FS13
c      write(59,*) FS21, ' ', FS22, ' ', FS23
c      write(59,*) FS31, ' ', FS32, ' ', FS33
c      write(59,*) ' '
c
      sig1(i) = 1./detF*(FS11*hsv(i,1)+FS12*hsv(i,4)+FS13*hsv(i,7))
      sig2(i) = 1./detF*(FS21*hsv(i,2)+FS22*hsv(i,5)+FS23*hsv(i,8))
      sig3(i) = 1./detF*(FS31*hsv(i,3)+FS32*hsv(i,6)+FS33*hsv(i,9))
      sig4(i) = 1./detF*(FS11*hsv(i,2)+FS12*hsv(i,5)+FS13*hsv(i,8))
      sig5(i) = 1./detF*(FS21*hsv(i,3)+FS22*hsv(i,6)+FS23*hsv(i,9))
      sig6(i) = 1./detF*(FS11*hsv(i,3)+FS12*hsv(i,6)+FS13*hsv(i,9))
c
c      write(59,*) 'Cauchy Stresses '
c      write(59,*) sig1(i), ' ', sig4(i), ' ', sig6(i)
c      write(59,*) sig4(i), ' ', sig2(i), ' ', sig5(i)
c      write(59,*) sig6(i), ' ', sig5(i), ' ', sig3(i)
c      write(59,*) ' '
c      write(59,*) '-----'
c
      endif
      return
      end

```

7.3. Изотропный билинейный материал с уравнением состояния

```

      subroutine umat44v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
* sig3,sig4,sig5,sig6,eps,hsvs,lft,llt,dtlsiz,capa,
* etype,tt,temps,failels,nlqa,crv,cma)
!      . etype,tt,temps,failels,nlqa,crv)
c
c*****
c|  Livermore Software Technology Corporation (LSTC) |
c|  ----- |
c|  Copyright 1987-2008 Livermore Software Tech. Corp |
c|  All rights reserved |
c*****
c
c  Изотропный с линейный упрочнением
c  Обновляется только девиатор, поэтому можно использовать с уравнением состояния
c
      parameter (third=1.0/3.0)

```

```

      include 'nlqparm'
c
c$omp threadprivate (/eosdloc/)
      common/eosdloc/pc(nlq)
c
      dimension cm(*),d1(*),d2(*),d3(*),d4(*),d5(*),d6(*),
* sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*),
* eps(*),hsvs(nlqa,*),dt1siz(*),temps(*),crv(lq1,2,*),cma(*),
* failels(*)
      character*5 etype
      logical failels
c модуль сдвига, напряжения текучести, упрочнение, максимальное? давление
      g =cm(1)
      sy0 =cm(2)
      h =cm(3)
      pcut=cm(4)
c предельные пластические деформации
      epsfail=cm(5)
c
      ofac=1.0/(3.0*g+h)
      twog=2.0*g
c
      do i=lft,llt
c девиатор упругих напряжений
      davg=third*(d1(i)+d2(i)+d3(i))
      savg=third*(sig1(i)+sig2(i)+sig3(i))
      sig1(i)=sig1(i)-savg+twog*(d1(i)-davg)
      sig2(i)=sig2(i)-savg+twog*(d2(i)-davg)
      sig3(i)=sig3(i)-savg+twog*(d3(i)-davg)
      sig4(i)=sig4(i)+g*d4(i)
      sig5(i)=sig5(i)+g*d5(i)
      sig6(i)=sig6(i)+g*d6(i)
c radial return
      aj2=sqrt(1.5*(sig1(i)**2+sig2(i)**2+sig3(i)**2)+
*          3.0*(sig4(i)**2+sig5(i)**2+sig6(i)**2))
      sy=sy0+h*eps(i)
      eps(i)=eps(i)+ofac*max(0.0,aj2-sy)
      synew=sy0+h*eps(i)
      scale=synew/max(synew,aj2)
c scaling for radial return
      sig1(i)=scale*sig1(i)
      sig2(i)=scale*sig2(i)
      sig3(i)=scale*sig3(i)
      sig4(i)=scale*sig4(i)
      sig5(i)=scale*sig5(i)
      sig6(i)=scale*sig6(i)
c set pressure cut-off
      pc(i)=pcut
c проверка на разрушение
      failels(i)=eps(i).gt.epsfail
c
      enddo
      return
      end

```

7.4. Резина (применение деформационного градиента)

```

      subroutine umat45 (cm,eps,sig,epsps,hsv,dtl,capa,
* etype,time,temp,failel,crv,cma)
c
c*****
c| Livermore Software Technology Corporation (LSTC) |
c| ----- |

```

```

c| Copyright 1987-2008 Livermore Software Tech. Corp. |
c| All rights reserved |
c*****
c
c      Материал нео-Гука (пример подпрограммы пользователя)
c
c переменные
c      cm(1)=модуль Юнга
c      cm(2)=коэффициент Пуассона
c
c      Прочие переменные аналогичны Примеру 1
c
      include 'nlqparm'
      include 'iounits.inc'
      include 'bk06.inc'
      character*5 etype
      dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*),cma(*)
      logical fail1
c
      if (ncycle.eq.1) then
          call usermsg('mat45')
      endif
c Вычисление параметров Ламэ
      xlambda=cm(1)*cm(2)/((1.+cm(2))*(1.-2.*cm(2)))
      xmu=.5*cm(1)/(1.+cm(2))
c
      if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
*         etype.eq.'sld2d'.or.etype.eq.'tshel') then
c деформационный градиент сохранён ранее в переменных истории hsv(1),...,hsv(9)
c Вычисление якобиана
          detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
*          -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
*          +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c compute left cauchy-green tensor
          b1=hsv(1)*hsv(1)+hsv(4)*hsv(4)+hsv(7)*hsv(7)
          b2=hsv(2)*hsv(2)+hsv(5)*hsv(5)+hsv(8)*hsv(8)
          b3=hsv(3)*hsv(3)+hsv(6)*hsv(6)+hsv(9)*hsv(9)
          b4=hsv(1)*hsv(2)+hsv(4)*hsv(5)+hsv(7)*hsv(8)
          b5=hsv(2)*hsv(3)+hsv(5)*hsv(6)+hsv(8)*hsv(9)
          b6=hsv(1)*hsv(3)+hsv(4)*hsv(6)+hsv(7)*hsv(9)
c ВЫЧИСЛИТЬ напряжения в форме Коши
          detfinv=1./detf
          dmu=xmu-xlambda*log(detf)
          sig(1)=detfinv*(xmu*b1-dmu)
          sig(2)=detfinv*(xmu*b2-dmu)
          sig(3)=detfinv*(xmu*b3-dmu)
          sig(4)=detfinv*xmu*b4
          sig(5)=detfinv*xmu*b5
          sig(6)=detfinv*xmu*b6
c
          else if (etype.eq.'shell') then
c деформационный градиент сохранён ранее в переменных истории hsv(1),...,hsv(9)
c compute part of left cauchy-green tensor
c independent of thickness strain increment
          b1=hsv(1)*hsv(1)+hsv(4)*hsv(4)+hsv(7)*hsv(7)
          b2=hsv(2)*hsv(2)+hsv(5)*hsv(5)+hsv(8)*hsv(8)
          b4=hsv(1)*hsv(2)+hsv(4)*hsv(5)+hsv(7)*hsv(8)
c secant iterations for zero normal stress
          do iter=1,5
c first thickness strain increment initial guess
c assuming Poisson's ratio different from zero
              if (iter.eq.1) then
                  eps(3)=-xlambda*(eps(1)+eps(2))/(xlambda+2.*xmu)
c second thickness strain increment initial guess
              else if (iter.eq.2) then

```



```

        sigold=sig(3)
        epsold=eps(3)
        eps(3)=0.
c secant update of thickness strain increment
        else if (abs(sig(3)-sigold).gt.0.0) then
            deps=- (eps(3)-epsold) / (sig(3)-sigold)*sig(3)
            sigold=sig(3)
            epsold=eps(3)
            eps(3)=eps(3)+deps
        endif
c compute last row of deformation gradient
        call compute_f3s(hsv(3),hsv(6),hsv(9),eps(3))
c ВЫЧИСЛИТЬ ЯКОБИАН
        detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
        *   -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
        *   +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c compute normal component of left cauchy-green tensor
        b3=hsv(3)*hsv(3)+hsv(6)*hsv(6)+hsv(9)*hsv(9)
c ВЫЧИСЛИТЬ НОРМАЛЬНЫЕ НАПРЯЖЕНИЯ
        detfinv=1./detf
        dmu=xmu-xlambda*log(detf)
        sig(1)=detfinv*(xmu*b1-dmu)
        sig(2)=detfinv*(xmu*b2-dmu)
        sig(3)=detfinv*(xmu*b3-dmu)
        sig(4)=detfinv*xmu*b4
c ВЫХОД В СЛУЧАЕ МАЛОСТИ НОРМАЛЬНЫХ НАПРЯЖЕНИЙ
        if (abs(sig(3)).le.1.e-5*
        *   (abs(sig(1))+abs(sig(2))+abs(sig(4)))) goto 10
        enddo
c compute remaining components of left cauchy-green tensor
10      b5=hsv(2)*hsv(3)+hsv(5)*hsv(6)+hsv(8)*hsv(9)
        b6=hsv(1)*hsv(3)+hsv(4)*hsv(6)+hsv(7)*hsv(9)
c ВЫЧИСЛИТЬ ОСТАВШИЕСЯ КОМПОНЕНТЫ ТЕНЗОРА НАПРЯЖЕНИЙ
        sig(5)=detfinv*xmu*b5
        sig(6)=detfinv*xmu*b6
c модель материала применима только к твёрдотельникам и оболочкам
        else
c          write(iotty,20) etype
c          write(iohsp,20) etype
c          write(iomsg,20) etype
c          call adios(2)
c          cerdat(1)=etype
c          call lsmsg(3,MSG_SOL+1151,ioall,ierdat,rerdat,cerdat,0)
        endif
c
c20    format(/
c      1 ' *** Error element type ',a,' can not be',
c      2 '          run with the current material model. ')
        return
        end
!
        subroutine umat45v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
        * sig3,sig4,sig5,sig6,epsps,hsvs,lft,llt,dtlsiz,capa,
        * etype,tt,temps,failels,nlqa,crv,cma)
c
c*****
c|  Livermore Software Technology Corporation  (LSTC)          |
c|  -----|
c|  Copyright 1987-2008 Livermore Software Tech. Corp        |
c|  All rights reserved                                       |
c*****
c
        include 'nlqparm'
        include 'nhisparm.inc'
c$omp threadprivate (/bk36loc/)

```

```

common/bk36loc/index
dimension d1(*),d2(*),d3(*),d4(*),d5(*),d6(*)
dimension sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*)
dimension cm(*),epsps(*),hsvs(nlq,*),dtlsiz(*)
dimension temps(*),crv(lq1,2,*),cma(*)
logical failels(*)
character*5 etype

C
dimension sig(6),eps(6),hsv(NHISVAR)
C
do i=lft,llt
  index=i
  sig(1)=sig1(i)
  sig(2)=sig2(i)
  sig(3)=sig3(i)
  sig(4)=sig4(i)
  sig(5)=sig5(i)
  sig(6)=sig6(i)
C
  eps(1)=d1(i)
  eps(2)=d2(i)
  eps(3)=d3(i)
  eps(4)=d4(i)
  eps(5)=d5(i)
  eps(6)=d6(i)
C
  hsv(1)=hsvs(i,1)
  hsv(2)=hsvs(i,2)
  hsv(3)=hsvs(i,3)
  hsv(4)=hsvs(i,4)
  hsv(5)=hsvs(i,5)
  hsv(6)=hsvs(i,6)
  hsv(7)=hsvs(i,7)
  hsv(8)=hsvs(i,8)
  hsv(9)=hsvs(i,9)
C
  call umat45(cm,eps,sig,epsps(i),hsv,dtlsiz(i),capa,etype,tt,
*   temps(i),failels(i),crv,cma)
C
  sig1(i)=sig(1)
  sig2(i)=sig(2)
  sig3(i)=sig(3)
  sig4(i)=sig(4)
  sig5(i)=sig(5)
  sig6(i)=sig(6)
C
  hsvs(i,1)=hsv(1)
  hsvs(i,2)=hsv(2)
  hsvs(i,3)=hsv(3)
  hsvs(i,4)=hsv(4)
  hsvs(i,5)=hsv(5)
  hsvs(i,6)=hsv(6)
  hsvs(i,7)=hsv(7)
  hsvs(i,8)=hsv(8)
  hsvs(i,9)=hsv(9)
  d3(i)=eps(3)
enddo
return
end

!
!
subroutine utan45(cm,eps,sig,epsps,hsv,dtl,capa,etype,tt,
* temper,es,crv,failel,cma)
C
C*****

```

```

c|  Livermore Software Technology Corporation   (LSTC)           |
c|  -----|
c|  Copyright 1987-2008 Livermore Software Tech. Corp           |
c|  All rights reserved                                         |
c*****
c
      include 'nlqparm'
      dimension cm(*),eps(*),sig(*),hsv(*)
      dimension es(6,*),crv(101,2,*),cma(*)
      logical failel
      character*5 etype
c без переменных истории, NHV=0
c деформационный градиент сохранён в hsv(1),...,hsv(9)
c ВЫЧИСЛИТЬ ЯКОБИАН
      detf=hsv(1)*(hsv(5)*hsv(9)-hsv(6)*hsv(8))
      *   -hsv(2)*(hsv(4)*hsv(9)-hsv(6)*hsv(7))
      *   +hsv(3)*(hsv(4)*hsv(8)-hsv(5)*hsv(7))
c ВЫЧИСЛИТЬ параметры Ламэ
      xlambd=cm(1)*cm(2)/((1.+cm(2))*(1.-2.*cm(2)))
      xmu=.5*cm(1)/(1.+cm(2))
c ВЫЧИСЛИТЬ модуль упрочнения (tangent stiffness)
c одинако для оболочек и твёрдотельников
      detfinv=1./detf
      dmu=xmu-xlambd*log(detf)
      es(1,1)=detfinv*(xlambd+2.*dmu)
      es(2,2)=detfinv*(xlambd+2.*dmu)
      es(3,3)=detfinv*(xlambd+2.*dmu)
      es(4,4)=detfinv*dmu
      es(5,5)=detfinv*dmu
      es(6,6)=detfinv*dmu
      es(2,1)=detfinv*xlambd
      es(3,2)=detfinv*xlambd
      es(3,1)=detfinv*xlambd
      es(1,2)=es(2,1)
      es(2,3)=es(3,2)
      es(1,3)=es(3,1)
      return
      end
!
      subroutine utan45v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
* sig3,sig4,sig5,sig6,epsps,hsvs,lft,llt,dtlsiz,capa,
* etype,tt,temps,dsave,nlqa,crv,failels,cma)
c
c*****
c|  Livermore Software Technology Corporation   (LSTC)           |
c|  -----|
c|  Copyright 1987-2008 Livermore Software Tech. Corp           |
c|  All rights reserved                                         |
c*****
c
      include 'nlqparm'
      include 'nhisparm.inc'
      dimension d1(*),d2(*),d3(*),d4(*),d5(*),d6(*)
      dimension sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*)
      dimension epsps(*),hsvs(nlq,*),dtlsiz(*),cm(*)
      dimension temps(*),dsave(nlq,6,*),crv(101,2,*),cma(*)
      logical failels(*)
      character*5 etype
c
      dimension sig(6),eps(6),hsv(NHISVAR),es(6,6)
      logical failel
c
      do i=lft,llt
         sig(1)=sig1(i)
         sig(2)=sig2(i)

```

```

      sig(3)=sig3(i)
      sig(4)=sig4(i)
      sig(5)=sig5(i)
      sig(6)=sig6(i)
c
      eps(1)=d1(i)
      eps(2)=d2(i)
      eps(3)=d3(i)
      eps(4)=d4(i)
      eps(5)=d5(i)
      eps(6)=d6(i)
c
      hsv(1)=hsvs(i,1)
      hsv(2)=hsvs(i,2)
      hsv(3)=hsvs(i,3)
      hsv(4)=hsvs(i,4)
      hsv(5)=hsvs(i,5)
      hsv(6)=hsvs(i,6)
      hsv(7)=hsvs(i,7)
      hsv(8)=hsvs(i,8)
      hsv(9)=hsvs(i,9)
c
      do k=1,6
        do j=1,6
          es(j,k)=0.
        enddo
      enddo
c
      failel=failels(i)
c
      call utan45(cm,eps,sig,epsps(i),hsv,dtlsiz(i),capa,etype,tt,
*      temps(i),es,crv,failel,cma)
c
      do k=1,6
        do j=1,6
          dsave(i,j,k)=es(j,k)
        enddo
      enddo
      enddo
      return
      end

```

7.5. Изотропный упругий для тепловых задач

```

      subroutine umat46 (cm,eps,sig,epsps,hsv,dtl,capa,etype,tt,
*      temper,failel,crv,cma,thhsvi,nthhsv)
c
c*****
c|  Livermore Software Technology Corporation  (LSTC)  |
c|  -----  |
c|  Copyright 1987-2008 Livermore Software Tech. Corp  |
c|  All rights reserved  |
c*****
c
c Изотропный упругий материала с доступом к переменным истории соответствующего
c теплового UM. См. также подпрограмму ЛУМ (Пример 1), где температура не
c учитывалась. Для доступа к тепловым переменным истории оболочек и
c твёрдотельников требуется установить флаг IHVE.eq.1 в ключевом слове
c *MAT_THERMAL_USER_DEFINED. Подробности в комментариях.
c
      include 'nlqparm'
      include 'bk06.inc'
      include 'iounits.inc'

```

```

      dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*),thhsvi(*),
*   sma(*)
      character*5 etype
      logical failel
C
C Переменные только для чтения
C
C   cm(1)=модуль Юнга
C   cm(2)=коэффициент Пуассона
C   cm(3)=.eq.0 ничего не делать
C           .eq.1 вывести в консоль механические и тепловые переменные истории.
C
C   nthhsv=количество тепловых переменных истории.
C   thhsvi=в ключевом слове *MAT_THERMAL_USER_DEFINED флаг INVE.eq.1, тогда
C           массив thhsvi(i), где i=1...nthhsv, содержит переменные истории
C           из процедуры теплового UM, интерполированные для текущей
C           точки интегрирования с использованием функции формы для
C           соответствующего теплового конечного элемента.
C   ВЫЧИСЛИТЬ модуль сдвига G
      if (ncycle.eq.1) then
        call usermsg('mat46')
      endif
C
C   g2 =cm(1)/(1.+cm(2))
C   g  =.5*g2
C
C   if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
*     etype.eq.'sld2d'.or.etype.eq.'tshel') then
      davg=(-eps(1)-eps(2)-eps(3))/3.
      p=-davg*cm(1)/(1.-2.*cm(2))
      sig(1)=sig(1)+p+g2*(eps(1)+davg)
      sig(2)=sig(2)+p+g2*(eps(2)+davg)
      sig(3)=sig(3)+p+g2*(eps(3)+davg)
      sig(4)=sig(4)+g*eps(4)
      sig(5)=sig(5)+g*eps(5)
      sig(6)=sig(6)+g*eps(6)
C
      else if (etype.eq.'shell') then
        gc   =capa*g
        q1   =cm(1)*cm(2)/((1.0+cm(2))*(1.0-2.0*cm(2)))
        q3   =1./(q1+g2)
        eps(3)=-q1*(eps(1)+eps(2))*q3
        davg  =(-eps(1)-eps(2)-eps(3))/3.
        p     =-davg*cm(1)/(1.-2.*cm(2))
        sig(1)=sig(1)+p+g2*(eps(1)+davg)
        sig(2)=sig(2)+p+g2*(eps(2)+davg)
        sig(3)=0.0
        sig(4)=sig(4)+g *eps(4)
        sig(5)=sig(5)+gc*eps(5)
        sig(6)=sig(6)+gc*eps(6)
C
      else
C   write(iotty,10) etype
C   write(iohsp,10) etype
C   write(iomsg,10) etype
C   call adios(2)
      cerdat(1)=etype
      call lsmmsg(3,MSG_SOL+1152,ioall,ierdat,rerdat,cerdat,0)
      endif
C   ВЫВОДИТЬ переменные истории в консоль?
      if ( nint(cm(3)).eq.1) then
        write(iotty,*) " epsp ",epsp," thhsvi(1) ",thhsvi(1),
*   " thhsvi(5) ",thhsvi(5),nthhsv
      endif
C

```

```

c10  format(/
c   1 ' *** Error element type ',a,' can not be',
c   2 '          run with the current user material model #47.')
```

return
end

!

subroutine umat46v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
* sig3,sig4,sig5,sig6,epsps,hsvs,lft,llt,dtlsiz,capa,
* etype,tt,temps,failels,nlqa,crv,cma,thhsv,nthhsv)

C
C*****
c| Livermore Software Technology Corporation (LSTC) |
c| ----- |
c| Copyright 1987-2008 Livermore Software Tech. Corp |
c| All rights reserved |
C*****
C

include 'nlqparm'
include 'nhisparm.inc'
dimension d1(*),d2(*),d3(*),d4(*),d5(*),d6(*)
dimension sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*)
dimension cm(*),epsps(*),hsvs(nlq,*),dtlsiz(*),thhsv(nlq,*),
dimension temps(*),crv(lq1,2,*),cma(*)
logical failels(*)
character*5 etype

C
c Подробности см. в подпрограмме umat46.
C

dimension sig(6),eps(6),hsv(NHISVAR),thhsvi(100)

C

do i=lft,llt
sig(1)=sig1(i)
sig(2)=sig2(i)
sig(3)=sig3(i)
sig(4)=sig4(i)
sig(5)=sig5(i)
sig(6)=sig6(i)
eps(1)=d1(i)
eps(2)=d2(i)
eps(3)=d3(i)
eps(4)=d4(i)
eps(5)=d5(i)
eps(6)=d6(i)

C

do j=1,nthhsv
thhsvi(j)=thhsv(i,j)
enddo

C

call umat46(cm,eps,sig,epsps(i),hsv,dtlsiz(i),capa,etype,tt,
* temps(i),failels(i),crv,cma,thhsvi,nthhsv)

C

do j=1,nthhsv
thhsv(i,j)=thhsvi(j)
enddo

C

sig1(i)=sig(1)
sig2(i)=sig(2)
sig3(i)=sig(3)
sig4(i)=sig(4)
sig5(i)=sig(5)
sig6(i)=sig(6)
d3(i)=eps(3)
enddo
return
end

7.6. Упруго-пластический материал по Мизесу и алгоритм интегрирования напряжений

Von Mises User Material along with the stress integration procedure with cutting plane algorithm. Подробнее об алгоритме см. [21], UM-код расположен по ссылке <https://sites.google.com/site/aenader/umat-workshop/umat-fortran>.

```

      subroutine umat43 (cm,eps,sig,epsp2,hsv,dt1,ca,pa,etype,tt,
1 temper,failel,crv)
!
! Copyright (C) 2005 NADER ABEDRABBO
! Данное ПО относится к категории free software: Вы можете распространять его
! и/или модифицировать на условиях GNU General Public License version 3
! или иной более свежей версии данной лицензии. Автор не несёт никакой
! ответственности за возможную некорректную работу данного ПО.
! Автор: Nader Abedrabbo, PhD
! E-Mail: aenader1@yahoo.com
! Дата создания: Сентябрь 22, 2005
! Дата редакции: 09/24/2010
!
! Пластическое течение по Мизесу (функция текучести Мизеса). Применяется метод
! секущих плоскостей (Cutting Plane algorithm, метод Гомори) для явного решателя
!
! Постановка:
!   1. Предполагается ПНС, т.е., sig13=sig23=sig33=0
!   2. Формулировка на основе теории течения
!   3. Интегрирование напряжений базируется на Euler Backward Theory
!   4. Using Tangent Cutting Back Theory
!   5. без учёта скоростей деформирования (Rate-Independent Formulation)
! Замечания:
!   1) В Ls-Dyna переменные hisv(1) & hisv(2) всегда фиксированы во всех UMAT's
!      Здесь:
!      hisv(1) = эквивалентная (результатирующая) пластическая деформация, epn
!      hisv(2) = полное гидростатическое давление, davg
!   2) В Dyna для элементов оболочек необходимо находить напряжения sig13 и
!      sig23. Они вычисляются из теории упругости.
! Обозначения:
!   epn      : эквивалентная пластическая деформация
!   epsp     : приращение пластической деформации
!   sigold   : предыдущее напряжённое состояние
!   strss    : обновлённые напряжения
!   toli     : погрешность для критерии сходимости
!   hflag    : флаг используемого закона упрочнения (1 или 2)
! *****
! Переменные:
!   cm(1)    = модуль Юнга, E
!   cm(2)    = коэффициент Пуассона
!   cm(3)    = выбор закона упрочнения
!              = 1 - степенной (Hollomon Power Law)
!   cm(4)    = 'K' коэффициент упрочнения в степенном законе
!   cm(5)    = 'n' показатель упрочнения в степенном законе
!   cm(9)    = объёмный модуль
!   cm(10)   = модуль сдвига
!   cm(11)   = флаг включения отладочных сообщений
!
!   eps(1)   = x -компонента приращения локальных деформаций
!   eps(2)   = y -компонента приращения локальных деформаций
!   eps(3)   = z -компонента приращения локальных деформаций
!   eps(4)   = xy-компонента приращения локальных деформаций
!              (приращение сдвиговой деформации eps(4) = 2*eps-xy)
!

```

```

!   sig(1) = x -компонента локальных напряжений (Accumulative)
!   Sig(2)...sig(6) = y,z,xу,yz,xz-компоненты локальных напряжений
!
!   hisv(1) = эквивалентная (результурующая) пластическая деф-ция при t=n, epn
!   hisv(2) = полное гидростатическое давление, davg
!
!   dt1 =текущий шаг по времени
!   capa =коэффициент коррекции поперечного сдвига
!   time = текущее время в задаче.
!
  IMPLICIT NONE
!
  include 'nlqparm'
  include 'iounits.inc'
  common/bk06/idmmy,iaddp,ifil,maxsiz,ncycle,time(2,30)
  REAL cm(*),eps(*),sig(*),hsv(*),crv(101,2,*), time
  INTEGER :: idmmy,iaddp,ifil,maxsiz,ncycle
  character*5 etype
  logical failel
!
  INTEGER :: i, j, iter, hflag, PR
  REAL    :: epn, ams1, ams2, ams3, G, G2
  REAL    :: capa, tt, temper, dt1, epsp2
  REAL    :: ym, nu, Kh, nh, davg
  REAL    :: toli, f1, f2, f3
  REAL, DIMENSION (3,3) :: cc
  REAL, DIMENSION (3)  :: Strss, sigold, deps, epsp
!
  common/khard/ams1,ams2,ams3,ym,nu,hflag
! Определим глобальные переменные и восстановим значения переменных истории
  ym    = cm(1)
  nu    = cm(2)
  hflag = NINT(cm(3))
  Kh    = cm(4)
  nh    = cm(5)
  PR    = NINT(cm(11)) ! флаг вывода сообщений
!
  toli  = 1.0E-6      ! точность итерационной процедуры
! инициализация матриц и массивов
  do i=1,3
    do j=1,3
      cc(i,j) = 0.0
    enddo !j
  enddo !i
! Вычисление модуля сдвига G
  G      = ym / (2.0*(1.0 + nu)) ! Modulus of Rigidity
  G2     = ym / (1.0 - nu*nu)   ! E/(1-v2)
! Сборка матрицы упругих модулей
  cc(1,1) = G2
  cc(1,2) = nu * G2
  cc(2,1) = nu * G2
  cc(2,2) = G2
  cc(3,3) = G      ! т.к. сдвиговая деформация 2*eps_xy else cc(4,4)=2*G
! Назначение величин коэффициентов упрочнения в зависимости от состояния
! флага hflag:
!   1 : степенной закон упрочнения (Power Law Hardening)
  if (hflag.eq.1) then
    ams1 = Kh
    ams2 = nh
    ams3 = 0.0
  else
    write(*,*)
    write(*,*) '*****'
    write(*,*) '*                E R R O R                *'
    write(*,*) '*****'
  end if

```



```

        write(*,*) '* FLAG ERROR: Check Flag Number For Hardening      *'
        write(*,*) '*                Model Type. In Input File, cm(3)    *'
        write(*,*) '*                Should be:                          *'
        write(*,*) '*                1: Power Law                               *'
        write(*,*) '*****'
        write(*,*)
        STOP
    endif
! Определим входные данные для основного кода
    deps(1) = eps(1)
    deps(2) = eps(2)
    deps(3) = eps(4)
    sigold(1) = sig(1)
    sigold(2) = sig(2)
    sigold(3) = sig(4)
    epn      = hsv(1)
! Подсчёт напряжений по теории течения
    call jaman(cc,deps,sigold,toli,epsp,epn,iter,Strss)
! Обновим итоговые напряжения
    sig(1) = Strss(1)
    sig(2) = Strss(2)
    sig(3) = 0.0
    sig(4) = Strss(3)
    sig(5) = sig(5) + capa*G*eps(5)
    sig(6) = sig(6) + capa*G*eps(6)
! Подсчёт полного гидростатического давления
    davg = - (1.0/3.0) * (sig(1)+sig(2)+sig(3))
! Вычисление приращения деформаций в направлении z
    f1 = epsp(1) + epsp(2)
    f2 = nu*(eps(1)+eps(2))
    f3 = 2.0*nu*f1
!
    eps(3) = (f1+f2-f3)/(nu-1.0)
    epsp2 = epn ! результирующая пластическая деформация, вернётся в модель
! Обновление переменных истории
    hsv(1) = epn
    hsv(2) = davg
    hsv(3) = iter
! Вывод в консоль отладочных сообщений. IFF (PR=1)
    if (PR.ne.1) PR = 0
    if (PR.eq.1) then
        write(*,*)
        write(*,*) '*****'
        write(*,*) '*                UMAT Using von Mises                *'
        write(*,*) ' Toli                = ', toli
        write(*,*) ' Youngs Modulus = ', ym
        write(*,*) ' HFLAG                = ', hflag
        write(*,*) ' ams1                = ', ams1
        write(*,*) ' ams2                = ', ams2
        write(*,*) ' ams3                = ', ams3
        write(*,*) '*****'
        write(*,*)
        STOP
    endif
    return
end

!
! *****
!
! subroutine jaman(cc,deps,sigold,toli,epsp,epn,iter,Strss)
! *****
! Автор: Nader Abedrabbo
! Дата : Март 25, 2003
! *****
! Данная подпрограмма корректно масштабирует напряжения и возвращает их
! на поверхность текучести. Входные аргументы: функция текучести, её производная,

```

```

! функция упрочнения. Метод секущих плоскостей возвращает напряжения
! на поверхность текучести. Выходные параметры: обновлённые напряжения,
! эквивалентная пластическая деформация, пластическая деформация, чисто итераций.
! *****
! Метод секущих плоскостей реализован на основе следующих источников:
! М. Ortiz and J.C. Simo, "An Analysis Of A New Class Of Integration
! Algorithm For Elastoplastic Constitutive Relations", Inter. Journal
! For Numerical Methods In Engineering, Vol.23, pp. 353-366, 1986.
! *****
!
REAL, Dimension (3,3):: cc
REAL, Dimension (3)  :: sigt, sigold, deps, epsp, epsp2
REAL, Dimension (3)  :: Strss, dphi, epspdot
REAL :: ep, epn, ams1, ams2, ams3
REAL :: f1, f2, f3, f4, ff1, epdot
REAL :: phi, hard2d, yldf, yld2, yldn, dlam, fac, kprim
REAL :: toli
INTEGER :: i, j, k, ii, jj, ik, hflag, iter
common/khard/ams1,ams2,ams3,ym,nu,hflag
!
ep   = epn
!
do i=1,3
    epsp2(i)  = 0.0
    sigt(i)   = sigold(i)
enddo !i
! Вычислить предварительные обновлённые полные напряжения (updated total trial
do i=1,3
    do j=1,3
        sigt(i) = sigt(i) + cc(i,j)*deps(j)
    enddo ! j
enddo ! I
! Вычислить предварительную функцию текучести (trial yield function) (at k)
call vmises(sigt,phi,dphi)
! Вычислить упрочняющуюся часть (Hardening part) (центр поверхности текучести)
call hardfunc(ep,hard2d,kprim)
! Проверка условия начала течения : упругость или пластика ?
yldf = phi - hard2d
!
if (phi.gt.hard2d) then
    iter = 0
    dlam = 0.0
!
100    fac = kprim
    do ii=1,3
        do jj=1,3
            fac = fac + dphi(ii)*cc(ii,jj)*dphi(jj)
        enddo !jj
    enddo !ii
    yld2 = phi - hard2d
    dlam = yld2 / fac
    ep   = ep + dlam
    ! новая эквивалентная пластическая деформация
    do ik = 1,3
        epspdot(ik)=dlam*dphi(ik)
        ! приращение пластической деформации
        epsp2(ik)=epsp2(ik)+epspdot(ik)
        ! полное приращение пласт. деф.
    enddo !ik
! Вычислить значения новых напряжений (cutting back)
do ii=1,3
    do jj=1,3
        sigt(ii) = sigt(ii) - dlam*cc(ii,jj)*dphi(jj)
    enddo !jj
enddo !ii
! Вычислить новую функцию текучести в соответствии с новыми напряжениями
call vmises(sigt,phi,dphi)
! Вычислить значение упрочнения из новой эквивалентной пласт. деформации

```

```

      call hardfunc(ep,hard2d,kprim)
! Проверка сходимости
      yldn    = phi - hard2d
      iter    = iter+1
!
      if (iter.gt.50) go to 110
      if (abs(yldn).gt.toli) go to 100
      go to 120
!
110    write(*,*)"Warning* Plasticity Algorithm Failed To Converge"
      write(*,1000)'|yldn| =',abs(yldn),'toli= ',toli,'Iter# =',iter
!
120    continue
!
      else                                ! Упругое состояние
          iter    = 0
      endif                                ! конец проверки пластичности
! Выведем в консоль обновленные значения напряжений, пластических деформаций,
! эквивалентную пластическую деформацию и число итераций
      epn    = ep
      do k = 1,3
          Strss(k) = sigt(k)
          epsp(k)  = epsp2(k)
      enddo !k
!
1000  Format (A8,f12.8,A10,f12.8,A12,I5)
      return
      end
!
! *****
! ****                                Процедура к функции Мизес                                ****
! *****
!
      subroutine vmises(sig,phi,dphi)
      implicit none
      REAL, Dimension (3)  :: sig, dphi
      REAL :: t1, t2, t3, phi
      INTEGER :: i
! Данная процедура вычисляет функцию текучести
! и её производную для функции Мизеса
      do i=1,3
          dphi(i) = 0.0
      enddo
!
      t1    = sig(1)
      t2    = sig(2)
      t3    = sig(3)
      phi   = sqrt(t1**2+t2**2-t1*t2+3.0*t3**2)
!
      dphi(1) = (2.0*sig(1) - sig(2)) / (2.0*phi)
      dphi(2) = (2.0*sig(2) - sig(1)) / (2.0*phi)
      dphi(3) = (3.0*sig(3)) / phi
      return
      end
!
! *****
! *****                                Законы упрочнения                                *****
! *****
!
      subroutine hardfunc(ep,hard2d,kprim)
      implicit none
      REAL :: ams1, ams2, ams3, fact, qh, ym, nu
      REAL :: ep, ee1, ee2, hard2d, kprim
      INTEGER :: hflag
      common/khard/ams1,ams2,ams3,ym,nu,hflag

```

```

! Вычисляет упрочняющаяся часть (hardening part); (центр поверхности текучести),
! Используется степенной закон упрочнения (Hollomon Power Law)
  if (hflag.eq.1) then
    fact      = 1.0/(ams2-1.0)
    qh        = (ym/ams1)**fact
    ee1       = (qh+ep)**ams2
    ee2       = (qh+ep)**(ams2-1.0)
    hard2d    = ams1 * ee1
    ! Isotropic hardening part dK/dep
    kprim     = ams1 * ams2 * ee2
  endif
  return
end

```

7.7. Полимер для термомеханических задач

```

c*****
  subroutine umat43 (cm,eps,sig,epsp,hsv,dtl,capa,etype,tt,
    1 temper,failel,crv)
c*****
c Molden polymer with Shear Rate and Temperature Dependent Viscosity
c
c   cm(3)=BULKMOD
c   cm(09)= CHTIME
c   cm(10)= N-EXP
c   cm(11)= A-EXP
c   cm(12)= MUINF
c   cm(13)= MU0 (изотермическая задача) или A (неизотермическая)
c   cm(14)= ACT-E
c   cm(15)= MUMAX
c   cm(16)= BULKVMIN
c
c   eps(1)...eps(6) =x,y,z,xy,yz,zx -компонента приращений локальных деформаций
c   sig(1)...sig(6)= x,y,z,xy,yz,zx -компонента полных напряжений
c
c   dtl=текущий шаг по времени
c   tt=текущее время в задаче
c   temper=текущая температура
c
c   hsv(16..21) = srg(mvi): градиент тензора скоростей касательных деформаций
c                   (Shear strain rate gradient)
c   hsv(22..27) = mu(mvi) : вязкость
c   hsv(28..33) = dst(mvi): компоненты девиатора тензора напряжений
c   hsv(34)     = phyd : давление
c   hsv(35..40) = sig(mvi): компоненты тензора полных напряжений
c
  include 'iounits.inc'
  character*5 etype
  dimension cm(*),eps(*),sig(*),hsv(*),crv(101,2,*)
  real dtl,temper,temp
  logical failel
c Локальные переменные
  dimension sg(6), srg(6), dst(6)
  real Rgas, mu(6), mub, phyd
  integer kron, mvi
c Газовая постоянная [J/(K*mol)]
  Rgas = 8.314510
c Нормальные компоненты тензора скоростей деформаций
  do 20 mvi=1,3
    if(dtl.le.0.or.eps(mvi).eq.0) then
      srg(mvi) = 0
    else

```

```

        srg(mvi) = eps(mvi)/dt1
        endif
        hsv(15+mvi)=srg(mvi)
20    continue
с Компоненты девиатора тензора скоростей деформаций
    do 25 mvi=4,6
        if(dt1.le.0.or.eps(mvi).eq.0) then
            srg(mvi) = 0
        else
            srg(mvi) = 0.5 * eps(mvi)/dt1
        endif
        hsv(15+mvi)=srg(mvi)
25    continue
с Вектор вязкостных компонент
    do 30 mvi=1,6
с неизотермический случай
        if (temper.ge.273) then
            mu(mvi) = cm(12) +
1            ( cm(13) * exp(cm(14)/(Rgas * temper)) - cm(12)) /
2            ( 1.0 + (cm(09) * abs(2*srg(mvi)) )**cm(11))
3            **((1-cm(10))/cm(11))
с изотермический случай
        else
            mu(mvi) = cm(12) +
1            ( cm(13) - cm(12) ) /
2            ( 1.0 + (cm(09) * abs(2*srg(mvi)) )**cm(11))
3            **((1-cm(10))/cm(11))
        endif
        if (mu(mvi).gt.cm(15)) then
            mu(mvi) = cm(15)
        endif
        hsv(21+mvi)=mu(mvi)
30    continue
с Объемная вязкость
        mub = - 2.0/3.0 * (mu(1) + mu(2) + mu(3))/3.0
        if (mub.lt.cm(16)) then
            mub = cm(16)
        endif
с Компоненты девиатора тензора напряжений (вектор)
        do 40 mvi=1,6
            if (mvi.le.3) then
                kron=1
            else
                kron=0
            endif
            dst(mvi) = 2 * mu(mvi) * srg(mvi) -
1            mub * (srg(1) + srg(2) + srg(3)) * kron
            hsv(27+mvi)=dst(mvi)
40    continue
с Давление (скаляр)
        phyd = -(sig(1) + sig(2) + sig(3))/3 -
1            cm(3) * (eps(1) + eps(2) + eps(3))
        hsv(34)=phyd
с Все компоненты тензора напряжений (вектор)
        do 70 mvi=1,6
            if (mvi.le.3) then
                kron=1
            else
                kron=0
            endif
            sig(mvi) = - phyd * kron + dst(mvi)
            hsv(34+mvi)=sig(mvi)
70    continue
        return
    end

```

7.8. Линейный упругий материал для неявного анализа

```

c Example provided by www.lsdyna-online.com
c
c Date: May, 2011
c
c E-Mail: courses@lsdyna-online.com
c
c Copyright, 2004 MPSE, Al Tabiei
c
c Copying for non-commercial usage allowed if copy bears this notice completely.
c
c NOTE: потребуется umat41 (для обновления напряжений) и utan41 (для формирования
c матрицы упругих констант es (material constitutive matrix))
c
      subroutine umat41 (cm,eps,sig,epsp,hsv,dt1,capa,etype,tt,
1 temper,failel,crv,cma)
      include 'nlqparm'
      include 'iounits.inc'
      common/bk06/idmmy,iaddp,ifil,maxsiz,ncycle,time(2,30)
      dimension cm(*),eps(*),sig(*),hsv(*),crv(101,2,*),cma(*)
      logical failel
      character*5 etype
c  ВЫЧИСЛИТЬ МОДУЛЬ СДВИГА, G
      g2 =cm(1)/(1.+cm(2))
      g  =.5*g2
c
      davg=(-eps(1)-eps(2)-eps(3))/3.
      p=-davg*cm(1)/(1.-2.*cm(2))
      sig(1)=sig(1)+p+g2*(eps(1)+davg)
      sig(2)=sig(2)+p+g2*(eps(2)+davg)
      sig(3)=sig(3)+p+g2*(eps(3)+davg)
      sig(4)=sig(4)+g*eps(4)
      sig(5)=sig(5)+g*eps(5)
      sig(6)=sig(6)+g*eps(6)
      return
      end
c
      subroutine utan41(cm,eps,sig,epsp,hsv,dt1,capa,etype,tt,
1 temper,es,crv,failel,cma)
      include 'nlqparm'
      dimension cm(*),eps(*),sig(*),hsv(*),crv(101,2,*),cma(*)
      dimension es(6,*)
      logical failel
      character*5 etype
c
      factor=1.
      if (failel) factor=1.e-8
c
      g=factor*.5*abs(cm(1))/(1.+cm(2))
      b=factor*abs(cm(1))/3./(1.-2.*cm(2))
      bg23=b-2.*g/3.
      bg43=b+4.*g/3.
c
      es(1,1)=bg43
      es(2,2)=bg43
      es(3,3)=bg43
      es(2,1)=bg23
      es(3,1)=bg23
      es(3,2)=bg23
      es(1,2)=es(2,1)
      es(1,3)=es(3,1)
      es(2,3)=es(3,2)
      es(4,4)=g

```

```

es(5,5)=g
es(6,6)=g
return
end

```

7.9. Упруго-пластический материал с изотропным упрочнением

```

c Fortran Program provided by Al Tabiei
c
c Date: Aug 27, 2004
c
c Tel: 513-3368075
c
c E-Mail: atabiei@aol.com
c
c Copyright, 2004 MPSE, Al Tabiei
c
c Copying for non-commercial usage allowed if copy bears this notice completely.
c
  subroutine umat41 (cm,eps,sig,epsp,hisv,dt1,capa,etype,tt,
1      temper,failel,crv)
  include 'iounits.inc'
  character*5 etype
  dimension cm(*),eps(*),sig(*),hisv(*),crv(101,2,*)
  logical failel
c
c      solid elastic-plastic material with isotropic hardening
c      A. Tabiei
c Переменные
c      cm(1)=модуль Юнга
c      cm(2)=коэффициент Пуассона
c      cm(3)=модуль сдвига
c      cm(4)=критические деформации (Hardening factor beta)
c      cm(5)=модуль упрочнения (Tangent modulus)
c      cm(6)=напряжения текучести
c      cm(7)=объёмный модуль
c
c      hisv(1)=полные пластические деформации
c      hisv(2)=полное гидростатическое давление
c
c      qh=модуль пластического упрочнения(plastic hardening modulus)
c      ak=напряжения текучести
c      q1-q6=компоненты девиатора тензора напряжений
c      ak2=функция текучести
c      depi=приращение пластической деформации
c      deps=scale back (radial return)
c
c      вычислить модуль сдвига, G
      g2 =cm(1) / (1.+cm(2))
      g  =.5*g2
      fs=cm(4)
c
      davg=(-eps(1)-eps(2)-eps(3))/3.
c      Вычисление гидростатического давления (приращение)
      p=-davg*cm(1)/((1.-2.*cm(2)))      ! 1
c      Вычисление полного гидростатического давления
      hisv(2)=hisv(2)-p
c      Вычисление предварительных напряжений (Trial Stress)
      sig(1)=sig(1)+p+g2*(eps(1)+davg)      ! 2
      sig(2)=sig(2)+p+g2*(eps(2)+davg)
      sig(3)=sig(3)+p+g2*(eps(3)+davg)
      sig(4)=sig(4)+g*eps(4)
      sig(5)=sig(5)+g*eps(5)

```

```

      sig(6)=sig(6)+g*eps(6)
      qh=cm(5)*cm(1)/(cm(1)-cm(5))
c Обновление напряжений текучести
      ak=cm(6)+qh*hisv(1)
c Подсчёт девиатора напряжений
      q1=hisv(2)+sig(1)           ! 3
      q2=hisv(2)+sig(2)
      q3=hisv(2)+sig(3)
      q4=sig(4)
      q5=sig(5)
      q6=sig(6)
      aj2=q4*q4+q5*q5+q6*q6+(q1*q1+q2*q2+q3*q3)/2
c Вычисление функции текучести
      ak2=3*aj2-ak*ak           ! 4
      scle=0
c Проверка текучести
      if(ak2.gt.0) then         ! 5
      scle=1
      endif
      fac1=1/(3.0*g+qh)
      fac2=3.0*g
      aj1=sqrt(3*abs(aj2))+1-scle
c Вычисление приращения пластических деформаций
      depi=scle*fac1*(aj1-ak)   ! 6
c Вычисление полных пластических деформаций
      hisv(1)=hisv(1)+depi
      epsp=hisv(1)
      deps=scle*fac2*depi/aj1
c Обновление напряжений
      sig(1)=sig(1)-deps*q1     ! 7
      sig(2)=sig(2)-deps*q2
      sig(3)=sig(3)-deps*q3
      sig(4)=sig(4)-deps*q4
      sig(5)=sig(5)-deps*q5
      sig(6)=sig(6)-deps*q6
c
      if (hisv(1).ge.fs) then
      failel=.true.
      sig(1)=0.0
      sig(2)=0.0
      sig(3)=0.0
      sig(4)=0.0
      sig(5)=0.0
      sig(6)=0.0
      endif
      return
      end

```

7.10. Композит с разрушением

```

c Fortran Program provided by Al Tabiei
c
c Date: Aug 27, 2004
c
c Tel: 513-3368075
c
c E-Mail: atabiei@aol.com
c
c Copyright, 2004 MPSE, Al Tabiei
c
c Copying for non-commercial usage allowed if copy bears this notice completely.
c

```



```

      subroutine umat41 (cm,eps,sig,epsp,hisv,dt1,capa,etype,tt,
1      temper,failel,crv)
      include 'iounits.inc'
      character*5 etype
      dimension cm(*),eps(*),sig(*),hisv(*),crv(101,2,*),c(3)
      logical failel
c A. Tabiei
c Инициализация
      e1=cm(1)
      e2=cm(2)
      v=cm(3)
      g=cm(4)
      Xt=cm(5)
      Yt=cm(6)
      Sm=cm(7)
      Yc=cm(8)
c
      b1=e1/(1.-v**2*e2/e1)
      b2=v*e2/(1.-v**2*e2/e1)
      b3=e2/(1.-v**2*e2/e1)
      b4=g
c
      c(1)=1.
      c(2)=1.
      c(3)=1.
c Hashin's Failure Prameters
      ffib=(sig(1)/Xt)**2+(sig(4)/Sm)**2
      ffibc=(sig(1)/Xt)**2
      fmax=(sig(2)/Yt)**2+(sig(4)/Sm)**2
      fcomp=(sig(2)/2/Sm)**2+(sig(4)/Sm)**2+
      .      ((Yc/2/Sm)**2-1)*sig(2)/Yc
c Hashin's Failure Criterion and Degredation
      if(fmax.gt. 1.0)then
      write(*,*)"matrix crack"
      c(2)=0.
      c(3)=0.
      endif
      if(sig(2).lt. 0.0 .and. fcomp.gt. 1.0)then
      write(*,*)"matrix compression"
      c(2)=0.
      endif
      if(ffibc.gt. 1.0 .or. (sig(1).gt. 0.0 .and. ffib.gt.Xt))then
      write(*,*)"fiber fracture"
      c(1)=0.
      c(2)=0.
      c(3)=0.
      endif
c
      hisv(1)=hisv(1)+eps(1)
      hisv(2)=hisv(2)+eps(2)
      hisv(3)=hisv(3)+eps(3)
c
      sig11a=b1*eps(1)-b2*eps(2)
      sig22a=b2*eps(1)+b3*eps(2)
      sig12a=b4*eps(4)
      sig23a=b4*capa*eps(5)
      sig13a=b4*capa*eps(6)
c
      sig(1)=(sig11a+sig(1))*c(1)
      sig(2)=(sig22a+sig(2))*c(2)
      sig(4)=(sig12a+sig(4))*c(3)
      sig(5)=sig12a+sig(5)
      sig(6)=sig12a+sig(6)
c
c      eps(3)=- (v/e1*(sig(1)-sig(1)))+

```

```

c      .      v/e2*(sig(2)-sig(2))
c
c      OPEN(1,FILE='out')
c      write(*,*)hisv(1),hisv(2),hisv(3)
c      return
c      end

```

7.11. Модель анизотропной пластичности Хилла

```

c
c      hill's normal anisotropic plasticity
c
c      common/aux2/d1(128),d2(128),d3(128),d4(128),d5(128),d6(128),
1 wzzdt(128),wyydt(128),wxzdt(128),einc(128)
c      common/aux11/po(128),
1 sign11(128),sig33s(128),da4(128),
2 da5(128),da6(128),t456(128),d1d2(128),da1(128),
3 da2(128),da3(128),deps(128),ak2(128),
4 aj2(128),depn(128),aks(128),scl1(128),scl2(128),depi(128),
5 t1(128),t2(128),t3(128),t4(128),t5(128),t6(128),ak(128),
6 epslst(128),epsnew(128),sg3new(128),sg3old(128),
7 sg3lst(128)
c      common/aux14/
1 sig1(128),sig2(128),sig3(128),sig4(128),
2 sig5(128),sig6(128), ep(128),epx1(128),epx2(128),
3 epx4(128),epx5(128),epx6(128),epx3(128),aux(128,43),
4 d7(128),fac1(128),fac2(128),g(128),
5 fac1qh(128),fac2qh(128),wq1(128),wq2(128),wq3(128),
6 wq4(128),blk(128),blkq(128),qh(128),ym(128)
c      common/aux18/dd(128),def(128)
c      common/aux19/
1 sign0(128),sign1(128),sign2(128),sign3(128),sign4(128),
2 sign5(128),sign6(128)
c      common/aux33/
1 ix1(128),ix2(128),ix3(128),ix4(128),ixs(128,4),mxt(128)
c      common/aux35/rhoa(128),cb(128),davg(128),p(128)
c      common/aux36/lft,llt
c      common/aux40/
1 a11(128),a12(128),a13(128),a21(128),a22(128),a23(128),
2 a31(128),a32(128),a33(128),s11(128),s12(128),s13(128),
3 s21(128),s22(128),s23(128),s31(128),s32(128),s33(128)
c      common/aux41/qq1(128),cbb(128),aj1(128)
c      common/tsarry/tslimt,tsarry(128)
c      common/sound/sndspd,sndsp(128),diagm(128),sarea(128),dxl(128)
c      common/hourg/yomod,gmod,ifsv
c      common/hour11/ebar(128),ebarmn(128),eyld(128),etanmd(128)
c      common/shlopt/istrn,istupd,ibelyt,miter
c      dimension cm(*)
c      dimension xl(128),phib2(128),xlnew(128),idone(128),
1 dep(128),xk2(128),xlold(128),phatold(128),phat(128)
c      dimension el1(128),el2(128),el3(128),error1(128)
c      data third/-.3333333333333333/
c
c      mx=48*(mxt(lft)-1)
c      r=cm(mx+21) !коэффициент анизотропии?
c      qs=cm(mx+11) !напряжения текучести?
c      pr=cm(mx+6) !коэффициент Пуассона
c      ss=cm(mx+2)
c      yomod=cm(mx+1) !модуль Юнга
c      gmod=.5*yomod/(1+pr) !модуль сдвига
c
c      fac3=pr/((1.0+pr)*(1.0-2.0*pr)) !константы для быстрого выполнения умножения
c      fac4=1./(1.+pr)

```

```

    fac5=-1./(1.-2.*pr)
    c=r/(1.+r)
    d=2.*(1.+2.*r)/(1.+r)
c
    dll=ymod/(1.-pr**2)           !скорость звука?
    sndspd=dll
    dl2=pr*dll
    d33=gmod
c
    do 5 i=lft,llt
        ym(i) =tsarry(i)*cm(mx+1)
        qh(i) =tsarry(i)*cm(mx+16)
        g(i)  =fac4*ym(i)           !модуль сдвига
        fac1(i)=1.5*g(i)
        blk(i) =fac5*ym(i)
        wq1(i) =fac3*ym(i)
        wq2(i) =.5*g(i)
        wq3(i) =1./(wq1(i)+2.0*wq2(i))
        wq4(i) =capa*wq2(i)
        eyl(i)=qs
        etanmd(i)=qh(i)
    5 continue
    if (ibelyt.eq.0) then
        do 25 i=lft,llt
            sign0(i)=sign3(i)+ym(i)*d3(i)
            sign3(i)=sign0(i)
    25 continue
    endif
    scl=0.
    do 30 i=lft,llt
        cb(i) =ss
        dld2(i)=d1(i)+d2(i)
        d3(i)=- (sig3(i)+wq1(i)*dld2(i))*wq3(i)
        da1(i)=sig1(i)+d11*d1(i)+d12*d2(i)
        da2(i)=sig2(i)+d12*d1(i)+d11*d2(i)
        da4(i)=sig4(i)+wq2(i)*d4(i)
        da5(i)=sig5(i)+wq4(i)*d5(i)
        da6(i)=sig6(i)+wq4(i)*d6(i)
        ak(i)  =qs+qh(i)*ep(i)
        aks(i) =ak(i)*ak(i)
    30 continue
    do 34 i=lft,llt
        aj2(i)=da1(i)**2 + da2(i)**2-2.*r/(r+1.)*da1(i)*da2(i)
    1      + 2.*(2.*r+1.)/(r+1.)*da4(i)**2
        ak2(i)=aj2(i)-aks(i)
        scle(i)=.50+sign(.5,ak2(i))
        scl1(i)=1.-scle(i)
        dep(i)=0.0
        einc(i)=d1(i)*sig1(i)+d2(i)*sig2(i)+d4(i)*sig4(i)+d5(i)*sig5(i)
    1      +d6(i)*sig6(i)
    34 continue
    do 40 i=lft,llt
    40 scl=scl+scl1(i)
    if (nint(scl).eq.0.) then
        do 46 i=lft,llt
            ebar(i)=ym(i)
            sig1(i)=da1(i)
            sig2(i)=da2(i)
            sig3(i)=0.
            sig4(i)=da4(i)
            sig5(i)=da5(i)
            sig6(i)=da6(i)
            einc(i)=d1(i)*sig1(i)+d2(i)*sig2(i)+d4(i)*sig4(i)+d5(i)*sig5(i)
    1      +d6(i)*sig6(i)+einc(i)
    46 continue

```

```

    go to 75
endif
xleft=0.0
right=2.0
tol=1.0e-6
do 60 i=lft,llt
  xlold(i)=xleft
  xl(i)=right
60 continue
call hilevl(xlold,ep,da1,da2,da4,pr,c,d,phatold,dep,ymod,
1   qs,qh,e11,e12,e13)
  iter=1
200 continue
call hilevl(xl,ep,da1,da2,da4,pr,c,d,phat,dep,
1   ymod,qs,qh,e11,e12,e13)
  do 70 i=lft,llt
    denom = phat(i)-phatold(i)
    if( scle(i) .ne. 0.0 ) then
      if( denom .eq. 0.0) stop 'denom 0 in hill'
      xlnew(i)=(xlold(i)*phat(i)-xl(i)*phatold(i))/denom
    endif
70 continue
  do 72 i=lft,llt
    error1(i)=abs(xlnew(i)-xl(i))/(abs(xl(i))+1.e-15)
    if( scle(i) .eq. 0.) then
      error1(i)=0.0
    endif
72 continue
c
  do 73 i=lft,llt
    idone(i)=0
    if( error1(i) .gt. tol) then
      xlold(i)=xl(i)
      phatold(i)=phat(i)
      xl(i)=xlnew(i)
      idone(i)=1
    endif
73 continue
  ifin=0
  do 80 i=lft,llt
    ifin=ifin+idone(i)
80 continue
  if(ifin.gt.0) then
    iter=iter+1
    go to 200
  endif
  do 90 i=lft,llt
    ebar(i)=qh(i)*scle(i)+ym(i)*(1.-scle(i))
    ep(i)=ep(i)+dep(i)
    sig1(i)=e11(i)*da1(i)+e12(i)*da2(i)
    sig2(i)=e12(i)*da1(i)+e11(i)*da2(i)
    sig3(i)=0.
    sig4(i)=e13(i)*da4(i)
    sig5(i)=da5(i)
    sig6(i)=da6(i)
    einc(i)=d1(i)*sig1(i)+d2(i)*sig2(i)+d4(i)*sig4(i)+d5(i)*sig5(i)
1   +d6(i)*sig6(i)+einc(i)
90 continue
  do 105 i=lft,llt
    d3(i)=((1.-pr**2)/ym(i)*xl(i)*(c-1.)-pr/ym(i))*(sig1(i)+sig2(i))
105 continue
75 return
end

```

8. Примеры расчётных файлов

Ниже представлены четыре примера простейших расчётных k-файлов и задания для самостоятельной переделки данных примеров.

8.1. Пружинный маятник с демпфером

Задание: конструкцию дополнить демпфером, пружинкой и массой; характеристики жёсткости, демпфирования и массы взять другими; соединить массы m_1 и m_2 стержнем, в центре стержня разместить третью массу; в момент времени $t=0$ приложить силу в одном из узлов стержня.

```

$ НИР Основы LS-DYNA Пример 1
$ (с) Казанцев А.В. 3 сент. 2017
$ email: alexkazancev@bk.ru
*KEYWORD
*TITLE
Pruzhynnyi mayatnik s dempferom
$
*PARAMETER
R massa          1.0
R jestkost       100.0
R dempfir        0.3
*NODE
  1, 0., 0., 0., 7,7
  2, 0., 0., -1.0, 4,7
*MAT_SPRING_ELASTIC
  1, &jestkost
*MAT_DAMPER_VISCOUS
  2, &dempfir
*SECTION_DISCRETE
1

*SECTION_DISCRETE
2

*PART
Spring
  1,1,1
Damper
  2,2,2
*ELEMENT_DISCRETE
  1,1,1,2
  2,2,1,2
*ELEMENT_MASS
  3,2, &massa
*INITIAL_VELOCITY_NODE
  2,0.0,0.0,1
*CONTROL_TERMINATION
  10.
*CONTROL_TIMESTEP
  ,0.01
*DATABASE_BINARY_D3PLOT
  0.01
*END

```

8.2. Плоский математический маятник

Задание: 1) преобразовать двумерный маятник в трёхмерный, по осях X и Y задать начальные скорости массы; 2) перейти к задаче о двойном плоском маятнике, добавив второй маятник к массе первого, маятники соединить с помощью шарнира (или иным способом).

```

$ НИР Основы LS-DYNA Пример 2
$ (с) Казанцев А.В. 2 сент. 2017
$ email: alexkazancev@bk.ru
*KEYWORD
*TITLE
Ploskii phizicheskii mayatnik
$
*PARAMETER
R massa      100.0
R jestkost   1.e05
R V          5.e-2
R L         -1.0
*NODE
  1, 0.0, 0.0, 0.0, 0, 0
  2, 0.0, 0.0, &L , 0, 0
*MAT_SPRING_ELASTIC
  1,&jestkost
*SECTION_DISCRETE
  1

*PART
Spring
  1,1,1
*ELEMENT_DISCRETE
  1,1,1,2
*ELEMENT_MASS
  2,2,&massa
*BOUNDARY_SPC
      1      0      1      1      1      1      0      1
      2      0      0      1      0      1      0      1
*INITIAL_VELOCITY_NODE
      2      &V      0.      0.      0.
*LOAD_BODY_Z
  1,9.81
*DEFINE_CURVE
  1
  0.0,0.0
  1.0,1.0
  10.0,1.0
*CONTROL_TERMINATION
  10.0
*CONTROL_TIMESTEP
  0.0,0.01

*DATABASE_BINARY_D3PLOT
  0.01

*END

```

8.3. Падение кубика на пластинку

Задание: преобразовать в задачу об отклике доски трамплина в бассейне, с которой спортсмены совершают прыжки (фактически, получить консольную балку): изменить размеры пластинки; разбить пластинку на четыре элемента; закрепить узлы одного ребра.

```

$ НИР Основы LS-DYNA Пример 3
$ (с) Казанцев А.В. 2 сент. 2017
$ email: alexkazancev@bk.ru
*KEYWORD
*TITLE
Padenie kubika na plastinku
*PARAMETER
$L1 - длина ребра кубика, L2 - длина грани пластины, H - расстояние между кубиком
и пластиной
$T - толщина пластинки
R L1          1.0
R L2          1.5
R H          -1.0
R T           0.1
*NODE
      1          0.          0.          0.
      2          &L1         0.          0.
      3          0.          &L1         0.
      4          &L1         &L1         0.
      5          0.          0.          &L1
      6          &L1         0.          &L1
      7          0.          &L1         &L1
      8          &L1         &L1         &L1
$
      9          &L2         &L2         &H
     10          &L2         0.          &H
     12          0.          0.          &H
     11          0.          &L2         &H
*MAT ELASTIC
  1,7850.,2.1e11,0.3
*ELEMENT_SOLID
  1      1      1      2      4      3      5      6      8      7
*ELEMENT_SHELL
  2      2      11     12     10     9
*SECTION_SOLID
  1,1
*SECTION_SHELL
      2      2
      &T      &T      &T      &T
$
*PART
Box
  1,1,1
*PART
Plate
  2,2,1
*SET_PART_LIST
  1
  1,2
$
*CONTACT_AUTOMATIC_GENERAL_ID
  0
  1,0,2

$ ГУ

```

```

*INITIAL_VELOCITY
1
      0.      0.     -1.0      0.      0.      0.
*SET_NODE_LIST
1
      8      6      7      5      1      3      4      2
$
*CONTROL_TERMINATION
1.5
*CONTROL_TIMESTEP
0., 0.1
*DATABASE_BINARY_D3PLOT
0.1
*END

```

8.4. Крутильный маятник

Задание: 1) найти формулу периода колебаний крутильного маятника в пособии Тимошенко С.П. Колебания в инженерном деле, 1985 г.; 2) в расчётном файле начальную скорость заменить на постоянно действующий крутящий момент; 3) преобразовать в задачу об определении НДС коленвала двухтактного двигателя: пусть дан коленвал с одним коленом и маховик, вращающиеся с угловой скоростью $\omega = 500$ рад/с; получить временной отклик НДС в конструкции в ответ на внезапное заклинивание поршня (т.е., закрепить узел шатунной шейки).

```

$ НИР Основы LS-DYNA Пример 4
$ (с) Казанцев А.В. 3 сент. 2017
$ email: alexkazancev@bk.ru
$
*KEYWORD
*TITLE
  Zakruchivanie vala
*PARAMETER
$ d1 - диаметр вала, d2 - диаметр диска
R d1      0.02
R d2      0.2
*NODE
1, 0.0, 0.0, 0.0
2, 0.0, 0.0,-1.0
3, 0.0, 0.0,-1.2
4,-0.1, 0.0,-1.0
*ELEMENT_BEAM
$ элемент 1 части 1, первый узел, второй узел, узел для указания третьей оси ЛСК
1, 1, 1, 2, 4
$ элемент 2, part 2, первый узел, второй узел, направление третьей оси ЛСК
2, 2, 2, 3, 4
*MAT_ELASTIC
$ создаём единственный материал
1,7850.0,2.1e11,0.3
*PART
$ часть 1, ей принадлежит конечный элемент со свойством 1 и материалом 1
Beam
1,1,1
$ часть 2, ей принадлежат конечный элемент со свойством 2 и материалом 1
Disk
2,2,1
*SECTION_BEAM
$ поперечное сечение вала
$ строка 1: секция № 1, 5-й параметр = 1 - трубчатое поперечное сечение
$ строка 2: два внешних диаметра в узлах, два внутренних диаметра

```



```
1,,,,,1
&d1, &d1, 0.0,0.0
*SECTION_BEAM_TITLE
Disk
2,,,,,1
&d2, &d2, 0.0,0.0
$
$ ГУ
$
*BOUNDARY_SPC_SET
$ закрепим все степени свободы узлового набора № 1
1,,1,1,1,1,1,1
*SET_NODE_LIST
$ строка 1: номер набора
$ строка 2: входящие в набор узлы
1
1
*INITIAL_VELOCITY_NODE
$ начальная угловая скорость  $\omega_z$  узлов № 2 и № 3
2,,,,,,0.05
3,,,,,,0.05
*CONTROL_TERMINATION
$ время расчёта
1.5
*CONTROL_TIMESTEP
$ для стабильности уменьшаем временной шаг в 20 раз
,0.05
*DATABASE_BINARY_D3PLOT
$ общие результаты сохранять каждые 0.01 сек
0.01
*DATABASE_HISTORY_BEAM
$ дополнительно записывать результаты для балок № 1 и № 2
1,2
$ вариант - использовать:
$ $*DATABASE_HISTORY_BEAM_ID
$ 1
$ 2
*DATABASE_BINARY_D3THDT
$ результаты для выбранных балок сохранять каждые 0.01 сек
0.01
*END
```