# Разработка и реализация для объемных конечных элементов модели композиционного материала с учетом прогрессирующего разрушения и динамического упрочнения

## Математическая формулировка модели прогрессирующего разрушения однонаправленного композиционного материала

На основе информации о моделях MAT\_161/162 (MAT\_COMPOSITE\_MSC) КПО по основным источникам [1-9] сформулирована математическая модель прогрессирующего разрушения однонаправленного композиционного материала. Модели материалов 161 и 162 в КПО могут использоваться для моделирования прогрессирующего разрушения в композитных материалах, состоящих из однонаправленных и тканых композитных лент, подвергнутых воздействию высокоинтенсивных нагрузок при больших скоростях деформаций. Критерии разрушения основаны на методологии, разработанной Хашином в 1980 году, при этом указанные модели могут использоваться для эффективного моделирования разрушения волокна, повреждения матрицы и расслоения при различных условиях разрушения. Модель с опцией DMG (материал 162) является обобщением модели разрушения основного слоя материала 161 путем принятия подхода механики поврежденных сред [2], в рамках которого после возникновения повреждения жесткостные характеристики материала уменьшаются по определенному закону.

Для однонаправленного композита имеем 6 критериев разрушения – 3 для волокон: при растяжении/сдвиге, при сжатии, при дроблении волокна по толщине слоя и 3 для матрицы: на плоскостях, параллельных волокнам, и для плоскости разрушения перпендикулярно и параллельно плоскостям расслоения. Для связи возникновения и роста повреждений с деградацией жесткостных свойств материала используется набор 6-ти переменных повреждений, которые участвуют в расчете элементов матрицы податливости, а по факту уменьшают значение модулей упругости и сдвига композита. При этом указанные переменные повреждений рассчитываются исходы из описанных выше критериев разрушения (мод) для однонаправленного и тканого композита. Для моделирования нелинейного поведения композита при прогрессирующем поведении используются 4 коэффициента для расчета поврежденности, исследованию значений указанных коэффициентов посвящена работа [5]. В разрабатываемой модели учитывается влияние скорости деформации на нелинейные характеристики напряженно-деформированного состояния композитного слоя, при этом используется логарифмическая зависимость от скорости деформации для определения модулей упругости и сдвига и прочностных параметров материала. При моделировании используются 4 коэффициента для учета зависимости значений прочностных свойств и жесткостных характеристик материала от скорости деформации. Модель включает в себя механизм удаления элементов (эрозию) при выполнении одного из трех условий.

Все критерии разрушения выражены в терминах компонентов напряжения, связанных с текущими деформациями в слое композита (ε1, ε2, ε3, ε12, ε23, ε31) = (εa, εb, εc, εab, εbc, εca), соответствующие модули упругости: (E1, E2, E3, G12, G23, G31) = (Ea, Eb, Ec, Gab, Gbc, Gca). Для рассматриваемой модели однонаправленного композита a, b и c обозначают поперечное и продольное направления в плоскости слоя и по толщине слоя соответственно;

Для однонаправленного композита имеем 6 критериев разрушения – 3 для волокон:

– при растяжении/сдвиге

, (1)

– при сжатии

, (2)

– при дроблении волокна по толщине слоя

 (3)

и 3 для матрицы:

– на плоскостях, параллельных волокнам

, (4)

– для плоскости разрушения перпендикулярно и параллельно плоскостям расслоения.

, (5)

, (6)

где 〈〉- скобки Маколея (неотрицательные значения либо нуль), и  – значения прочности на растяжение и сжатие в направлении волокна, а  и  – значения прочности слоя, связанные с разрывом и сдвигом волокна, соответственно,  и  - поперечные значения прочности на растяжение соответствующих режимов растяжения ( или );  – квазистатические значения прочности на сдвиг. При поперечной деформации сжатия ( или ) считается, что значения прочности на сдвиг зависят от нормальных деформаций сжатия, основанных на теории Мора-Кулона, т.е.

 (7)

где - материальная постоянная, так как аналогичен коэффициенту трения.

Соотношения (1)-(6) определяют шесть поверхностей поврежденности, параметры поврежденности  – радиусы поверхностей поврежденности имеют начальные значения равные 1 (нулевая поврежденность) и растут по мере возникновения и роста повреждений в соответствующих режимах.

Набор 6-ти переменных повреждений  используется при моделировании деградации жесткостных свойств материала для вычисления значений параметров поврежденности, влияющих на значения элементов матрицы податливости ( в процессе деградации по факту уменьшаются значения модулей упругости и сдвига композита).

Деградация свойств материала происходит в зависимости от 6-ти значений параметров поврежденности

 (8)

В начальный момент времени все значении параметров поврежденности равны нулю, значение поврежденности, как правило, не превышает 0,999 (входной параметр), значения  (АМ) являются входными параметрами модели и влияют на скорость поврежденности.

Для связи параметров поврежденности с описанными выше критериями (поверхностями) разрушения и значениями для однонаправленного материала имеем матрицу [q]

 (9)

Указанная матрица [q] позволяет учесть связь между параметрами и .

На рисунке 100 показана диаграмма влияния компонент тензора деформаций на модули материала, построенная на основании приведённых выше уравнений. Видно, что модель достаточно сложно учитывает взаимное влияние различных мод разрушения на деградацию отдельных модулей материала. Для нормальных компонент отдельно выделены деформации растяжения *Р* и сжатия *С*.

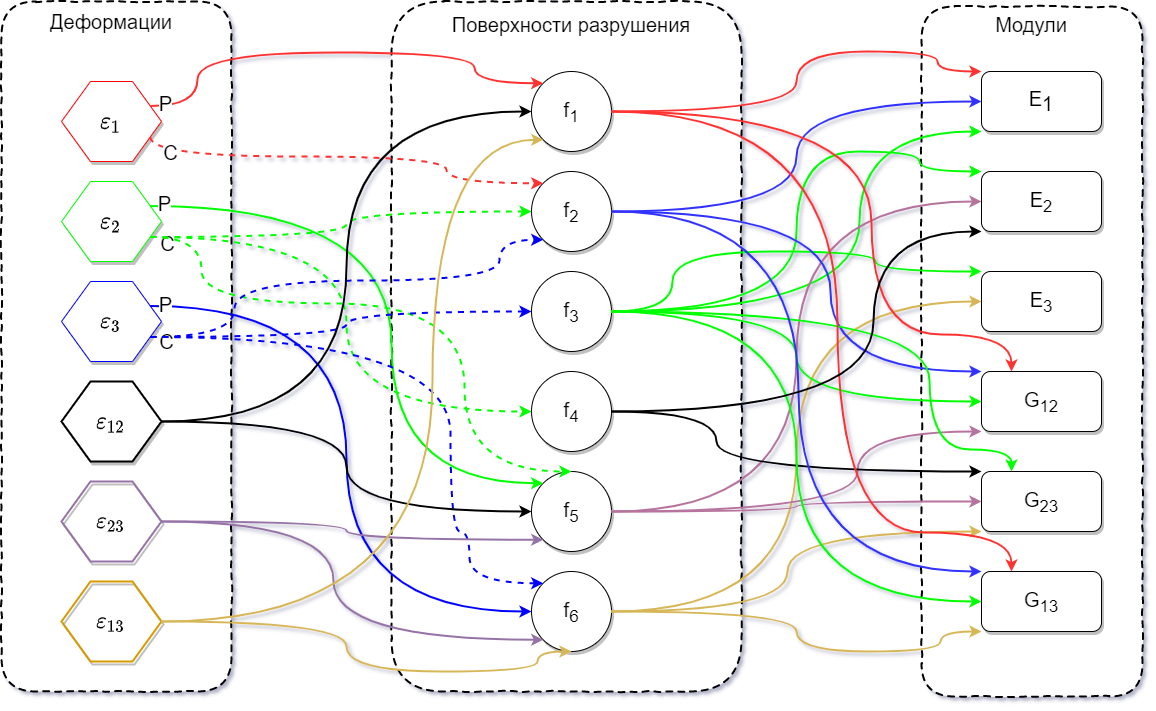


Рисунок 100 – Диаграмма влияния компонент тензора деформаций на модули материала

Влияние скорости деформации на нелинейную характеристику напряженно-деформированного состояния композитного слоя моделируется логарифмической зависимостью от скорости деформации для модулей упругости и сдвига и прочностные характеристики по формуле:

 (10)

где  – зависящее от скорости характеристика материала при средней скорости деформации , а  – квазистатическое значение соответствующей характеристики материала при эталонной (опорной) скорости деформации . В модели эталонная скорость деформации выбрана равной (11):

. (11)

Таким образом, единицей времени при использовании разрабатываемой модели должны быть секунда, иначе механизм учета влияния скоростей деформаций будет неэффективным.

*Влияние скорости деформации на прочностные свойства*

Первый параметр влияния скорости деформации на процесс разрушения Crate1 используется для учета эффектов скорости деформаций на прочностные свойства следующим образом:

 (12)

где векторы прочности и скорости деформации заданы соотношениями (13).

,  (13)

При этом в рассматриваемой модели полагается, что прочность на растяжение по толщине ScT и прочность на сдвиг Sab, Sbc и Sca не зависят от скорости деформаций.

*Влияние скорости деформации на жесткостные характеристики*

Три параметра учета скорости деформаций Crate2, Crate3 и Crate4 используются для добавления эффектов влияния скорости на три модуля упругости и три модуля сдвига в соответствии с соотношениями (14) и (15):

 (14)

, ,  (15)

Заметим, что влияние скорости оба модуля упругости в плоскости Ea и Eb управляются параметром скорости Crate2, а для модуля упругости по толщине Eс влияние оказывает параметр Сrate4. При этом влияние скорости деформации на модули сдвига Gab Gbcи Gca контролируется параметром Crate3.

В работе [6] приводятся результаты исследования свойств материала рассматриваемой модели, зависящих от скорости деформации.

Формирование матрицы податливости осуществляется из соотношений (16):

 (16)

Матрица жесткости [C] получается инвертированием матрицы податливости, т. е. .

Расчет напряжений осуществляется из соотношения для композитного ортотропного материала (17):

 (17)

Модель включает в себя механизм удаления элементов (эрозию) при выполнении любого из трех нижеприведенных условий:

1. Если в элементе однонаправленного композита осевая деформация растяжения превышает предельное значение E\_LIMIT.

2. Если в элементе превышено пороговое значение относительного объема сжатия (отношение текущего объема к первоначальному объему), т.е. для критерия разрушения материала сжимающий относительный объем (отношение текущего объема к исходному объему) в отказавшем элементе меньше, чем раздавливающий ECRSH.

3. Если в элементе превышено пороговое значение относительного объема расширения (отношение текущего объема к первоначальному объему), т.е. величина относительного объема при расширении в элементе больше, чем заданная величина EEXPN.

Таким образом, математическая модель прогрессирующего разрушения однонаправленного композита включает в себя соотношения (1)-(17) и три критерия удаления элемента, которые полностью описывают процесс деградации свойств однонаправленного композита с учетом влияния скорости деформации.

## Алгоритмы и блок-схемы программного модуля

Разработан алгоритм, позволяющий выполнить программную реализацию модели, аналогичной MAT\_COMPOSITE\_MSC\_DMG,

Алгоритм включает в себя следующие шаги:

1. Присвоение значений параметров материала по данным входного массива *Cmr* (всего разрабатываемая модель использует значения 35-ти параметров материала. Описание параметров модели содержится в Таблице 2 параграфа 1.3)
2. Расчет значений «парных» коэффициентов Пуассона. Используются известные соотношения для ортотропного материала –  и т.д., то есть по введенным трем значениям коэффициентов Пуассона и значениям модулей упругости материала определяются еще три коэффициента.
3. Расчет скорости звука в материале для определения величины шага по времени. Значение временного шага определяется стандартным образом и не пересчитывается в ходе расчетов.
4. Переход из глобальной в локальную систему координат. Используются преобразования, имеющиеся в ПП ЛОГОС.
5. Расчет приращений деформаций на текущем временном шаге.
6. Расчет полных текущих деформаций.
7. Проверка 3-х критериев удаления элемента (сравнение вычисленных на временном шаге значений соответствующих величин с заданными пороговыми значениями осевой деформации растяжения, относительного объема сжатия и расширения). Все критерии проверяются последовательно, для расчета относительного объема на каждом временном шаге вычисляется текущий объем элемента , который соотносится с первоначальным объемом. Элемент подлежит удалению (эрозия элемента), если выполняется хотя бы один из критериев.
8. В случае выполнения одного из критериев – удаление элемента (присвоение выходному параметру флаг-признак удаления элемента значения 1).
9. Корректировка механических и прочностных характеристик материала с учетом скорости деформации.
10. В зависимости от значения флага-признака материала модели (однонаправленный или тканый композит) расчет текущих радиусов 6-ти или 7-ми поверхностей поврежденности. На текущем этапе реализованы поверхности для однонаправленного композита.
11. Расчет шести параметров поврежденности с использованием текущих радиусов поверхностей разрушения и матрицы влияния *q*.
12. Формирование матрицы податливости с учетом поврежденности.
13. Расчет матрицы жесткости (обращение матрицы податливости).
14. Расчет напряжений.
15. Перевод напряжений в глобальную систему координат.

Описанный алгоритм ориентирован на то, что программный модуль будет разработан в рамках технологии интеграции пользовательского материала пакета программ ЛОГОС.

Блок-схема разработанного модуля представлена на рис. 1



Рис. 1 – Блок-схема программного модуля

## Формулировка требований к заданию параметров модели в ЛОГОС-ПреПост

На основе разработанной модели прогрессирующего разрушения композитного материала при динамическом нагружении сформулированы требования к заданию параметров модели в ЛОГОС-ПреПост, необходимые для реализация пользовательских интерфейсов.

Ниже приведена таблица 1, содержащая описание базовых параметров пользовательской модели материала (данные параметры должны быть заданы при использовании пользовательской модели материала в ПП ЛОГОС)

Таблица 1 – Базовые параметры пользовательской модели материала в ПП ЛОГОС

| **Наименование параметра** | **Функциональное назначение** | |
| --- | --- | --- |
| MID | Индекс пользовательской модели материала | |
| RO | Плотность | |
| MT | Явно в модели не используется  **Примечание: значение 0** | |
| LMC | Длина массива констант материала, которая равна числу констант, которые будут вводиться  **Примечание: для реализуемой модели материала это 35.** | |
| NHV | Число переменных истории нагружения  **Примечание: используется 15 параметров состояния, рассчитываемых внутри программного модуля)**  **1-6 - полные деформации по направлениям осей 1, 2, 3 и в плоскостях 12. 23, 13.**  **7-15 - параметры поврежденности (ось 1, 2, 3 - растяжение, ось 1, 2, 3 - сжатие, сдвиги в плоскостях 12, 23. 13)** | |
| IORTHO | 1, если материал ортотропный/анизотропный  **Примечание: задается значение 1** | |
| IBULK | Адрес модуля всестороннего сжатия в массиве констант материала  **Примечание: значение 0** | |
| IG | Адрес модуля сдвига в массиве констант материала  **Примечание: значение 0** | |
| IVECT | Явно в модели не используется  **Примечание: значение 0** | |
| IFAIL | Флаг-признак использования критериев разрушения  **Примечание: значение 0** | |
| ITHERMAL | Флаг-признак использования температуры. Вычисляется температура в элементе  **Примечание: значение 0** | |
| IHYPER | Флаг-признак гиперупругого материала  **Примечание: значение 0** | |
| IEOS | Флаг-признак использования уравнения состояния  **Примечание: значение 0** | |
| LMCA | Явно в модели не используется  **Примечание: значение 0** | |
| AOPT  (при условии IORTHO = 1) | Оси в материале:  = 0.0 локально ортотропный материал, в котором оси определены узлами 1,2 и 4 в элементе  = 1.0 локально ортотропный материал, в котором оси определены точкой в пространстве и положением центра элемента в глобальной системе координат. Используется только для объемных элементов  = 2.0 глобальная система координат  = 3.0 локально ортотропный материал, в котором оси определены поворотом осей вокруг нормали элемента на угол BETA от линии на плоскости элемента, заданной произведением вектора v и нормали элемента  = 4.0 локально ортотропный материал в цилиндрической системе координат с осями, определенными вектором v и исходной точкой P, которые задают продольную ось. Используется только для объемных элементов.  **Примечание: значение 0** | |
| MAXC  (при условии IORTHO = 1) | Флаг-признак смены осей в прямоугольных элементах:  = 1.0 по умолчанию  = 2.0 меняются оси а и b  = 3.0 меняются оси a и c  **Примечание: значение 1** | |
| XP YP ZP | Координаты точки p для опции AOPT = 1 (при условии IOPTHO = 1)  **Примечание: значения 0.0, 0.0, 0.0** | |
| A1 A2 A3 | Компоненты вектора а для опции AOPT = 2 (при условии IOPTHO = 1)  **Примечание: значения 0.0, 0.0, 0.0** | |
| V1 V2 V3 | Компоненты вектора v для опции AOPT = 3 (при условии IOPTHO = 1)  **Примечание: значения 0.0, 0.0, 0.0** | |
| D1 D2 D3 | Компоненты вектора d для опции AOPT = 2 (при условии IOPTHO = 1)  **Значения 0.0, 0.0, 0.0** | |
| BETA | Угол BETA в градусах для опции AOPT = 3 (при условии IOPTHO = 1)  **Примечание: значение 0** |
| IEVTS | Явно в модели не используется  **Примечание: значение 15** |

### Входные параметры модели (описание массива Cmr)

Модель использует значения 35-ти параметров – Таблица 2, это жесткостные и прочностные характеристики однонаправленного или тканого композита, а также ряд коэффициентов, влияющих на процесс деградации жесткостных свойств и управляющих влиянием скорости деформации на жесткостные и прочностные свойства композитного материала. В разрабатывемом программном модуле все указанные параметры содержатся в массиве Сmr.

Таблица 2 – Входные параметры модели (описание массива Cmr)

|  |  |
| --- | --- |
| **ДАЛЕЕ ИДУТ ПАРАМЕТРЫ МОДЕЛИ МАТЕРИАЛА - всего их 35** | |
| E1 | Модуль упругости в направлении оси 1 |
| E2 | Модуль упругости в направлении оси 2 |
| E3 | Модуль упругости в направлении оси 3 |
| PR21 | Коэффициент Пуассона 21 |
| PR31 | Коэффициент Пуассона 31 |
| PR32 | Коэффициент Пуассона 32 |
| G12 | Модуль сдвига 12 |
| G23 | Модуль сдвига 23 |
| G31 | Модуль сдвига 31 |
| X1T | Предельная прочность на растяжение в направлении оси 1 |
| X1C | Предельная прочность на сжатие в направлении оси 1 |
| X2T | Предельная прочность на растяжение в направлении оси 2 |
| X2C | Предельная прочность на сжатие в направлении оси 2 |
| X3T | Предельная прочность на растяжение в направлении оси 3 |
| SFC | Предельная прочность на раздавливание (прочность слоя, связанного с разрывом волокна) |
| SFS | Предельная прочность на сдвиг (прочность слоя, связанного со сдвигом волокна) |
| S12 | Квазистатическое значение прочности на сдвиг в плоскости 12 |
| S23 | Квазистатическое значение прочности на сдвиг в плоскости 23 |
| S31 | Квазистатическое значение прочности на сдвиг в плоскости 31 |
| SFFC | Масштабный коэффициент для остаточной прочности на сжатие |
| AMODEL | Флаг- признак материала модели  1 - модель однонаправленного композита  2 - модель тканевого (плетеного) композита |
| PHIC | Угол кулоновского трения для учета разрушения матрицы при расслаивании, значение от 0 до 90 |
| E\_LIMT | Пороговое значение осевой деформации растяжения для критерия разрушения материала (для тканого композита проверяются значения двух деформация растяжения в плоскости 12) |
| SDELM | Масштабный коэффициент для критерия расслоения |
| OMGMX | Предельное значение поврежденности для уменьшения модулей упругости (как правило, 0,999) |
| ECRSH | Пороговое значение относительного объема сжатия (отношение текущего объема к первоначальному объему) для критерия разрушения материала |
| EEXPN | Пороговое значение относительного объема расширения (отношение текущего объема к первоначальному объему) для критерия разрушения материала |
| CERATE1 | Коэффициент для учета зависимости значений прочностных свойств материала от скорости деформации |
| AM1 | Коэффициент для расчета поврежденности волокна в направлении 1 |
| AM2 | Коэффициент для расчета поврежденности матрицы при поперечном сжатии в направлении 2 (однонаправленный композит) или волокна в направлении 2 (тканый композит) |
| AM3 | Коэффициент для расчета поврежденности волокна при сдвиге в плоскости 12 |
| AM4 | Коэффициент для расчета поврежденности матрицы при расслаивании (направления 23 и 31) |
| CERATE2 | Коэффициент для учета зависимости значений модулей упругости в направлении координатных осей от скорости деформации |
| CERATE3 | Коэффициент для учета зависимости значений межслоевых модулей сдвига от скорости деформации |
| CERATE4 | Коэффициент для учета зависимости значения трансверсального модуля сдвига от скорости деформации |

В таблице 3, приведенной в [1], содержатся четыре набора значений параметров модели (с третьей строки после заголовка находятся все 35 параметров в несколько иной последовательности).

Таблица 3 – Значения параметров пользовательской модели материала [1]



### Структура хранения параметров состояния материала (описание массива *loc*)

Для программной реализации модели наряду с вышеописанными входными параметрами необходимо использование параметров состояния материла – Таблица 4.

Таблица 4 – Параметры состояния материала (описание массива *loc*)

| **Индекс loc** | **Наименование параметра** | **Функциональное назначение** |
| --- | --- | --- |
| 1 | w1 | Параметр поврежденности 1 |
| 2 | w2 | Параметр поврежденности 2 |
| 3 | w3 | Параметр поврежденности 3 |
| 4 | w4 | Параметр поврежденности 4 |
| 5 | w5 | Параметр поврежденности 5 |
| 6 | w6 | Параметр поврежденности 6 |
| 7 | r1 | Текущий радиус поверхности разрушения 1 |
| 8 | r2 | Текущий радиус поверхности разрушения 2 |
| 9 | r3 | Текущий радиус поверхности разрушения 3 |
| 10 | r4 | Текущий радиус поверхности разрушения 4 |
| 11 | r5 | Текущий радиус поверхности разрушения 5 |
| 12 | r6 | Текущий радиус поверхности разрушения 6 |
| 13 | r7 | Текущий радиус поверхности разрушения 7 |
| 14 | r8 | Текущий радиус поверхности разрушения 8 |
| 15 | r9 | Текущий радиус поверхности разрушения 9 |
| 16 | r10 | Текущий радиус поверхности разрушения 10 |
| 17 | r11 | Текущий радиус поверхности разрушения 11 |
| 18 | r12 | Текущий радиус поверхности разрушения 12 |
| 19 | r13 | Текущий радиус поверхности разрушения 13 |
| 20 | RS1 | Модуль ветви для остаточной прочности при сжатии в направлении A |
| 21 | RS2 | Модуль ветви для остаточной прочности при сжатии в направлении B |

Текст программы, реализующей описанные выше алгоритмы в рамках пользовательского материала ПП ЛОГОС, приводится в приложении 1.

## Программный макет модели

Программный макет реализуемой модели композиционного материала представляет собой код, написанный на языке *python* ([www.python.org](http://www.python.org)). Структура входных и выходных данных созданного макета максимально приближена к структуре данных модуля, реализующего пользовательский материал для ПП ЛОГОС-прочность. В качестве входных параметров процедура получает приращение компонент деформации на текущем шаге в виде массива [Δεx, Δεy, Δεz, Δεxy, Δεxz, Δεyz], массив параметров материала *Cmr*, массив напряжений с предыдущего расчетного шага [σx, σy, σz, τxy, τxz, τyz]. Остальные параметры, необходимые для модели, хранятся в массиве *loc* (параметры состояния материала или в терминах ПП ЛОГОС – исторические переменные). В текущем макете в данном массиве хранятся параметры поврежденности, используемые для корректировки компонент матрицы податливости, текущие радиусы поверхностей разрушения, а также параметры, необходимые для работы алгоритмов, реализующих остаточную прочность материала. В результате работы функция рассчитывает и возвращает значения напряжений на текущем расчетном шаге [σx, σy, σz, τxy, τxz, τyz].

Программный код, реализующий текущее состояние макета модели, приводится в приложении 2. Он позволяет определять компоненты тензора напряжений в условиях заданного деформированного состояния, либо в условиях одномерных напряженных состояний.

### Структура макета

Последовательность операций, выполняемых внутри процедуры, реализующей создаваемую модель материала, показана на схеме рисунка 500.

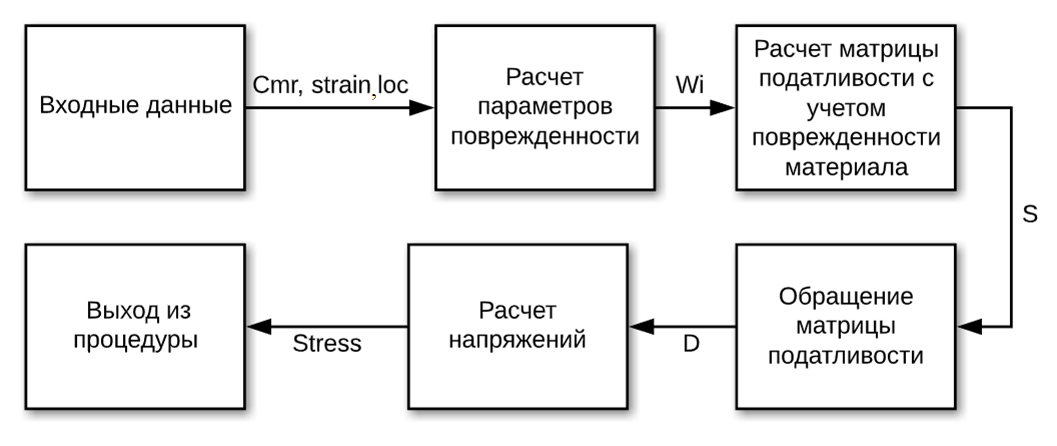


Рисунок 500 – Алгоритмическая схема макета

1. На основании входных данных: деформаций на текущем расчетном шаге *strain*, параметров состояния с предыдущего шага *loc*, и материальных констант *Cmr* рассчитываются параметры поврежденности материала
2. C учетом поврежденности определяется текущая матрица податливости
3. Путем обращения матрицы податливости определяется матрица жесткости поврежденного материала.
4. С использованием определяющих соотношений ортотропного материала рассчитываются напряжения

### Расчет поврежденности материала в макете

Ключевой особенностью реализуемой модели композиционного материала является присутствие в ней нескольких поверхностей разрушения, которые описывают отдельные режимы разрушения и деградации свойств армирующего волокна или ткани и матрицы.

За отчетный период на макете реализованы и протестированы все поверхности разрушения композиционного материала одномерного армирования: три для режимов разрушения армирующего волокна и три – для режимов разрушения матрицы.

В общем виде указанные поверхности можно записать в виде:



здесь *Cmr* – параметры материала, *strain* – вектор деформации.

Общий алгоритм оценки развития поврежденности в материале приведен на следующей схеме (рисунок 501):

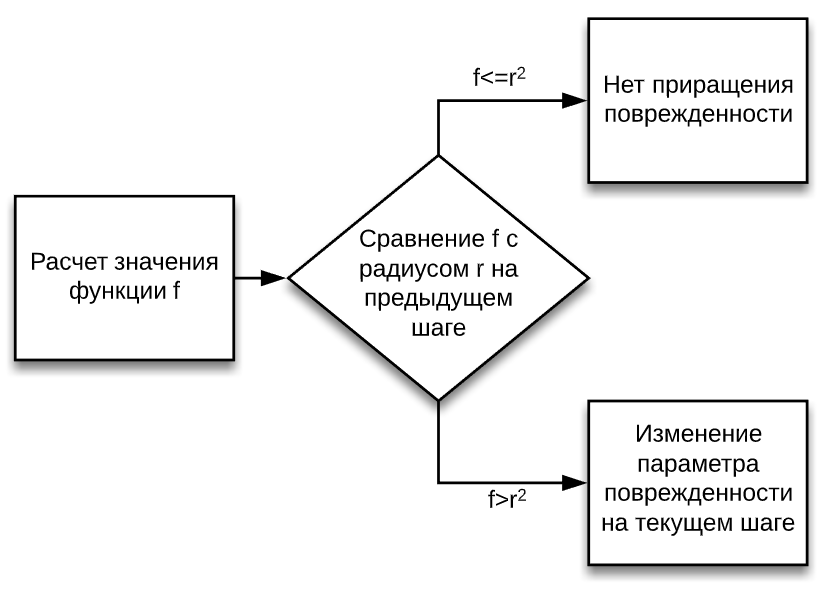


Рисунок 501 – Алгоритм расчета поврежденности

1. В начале каждого расчетного шага на основании поступивших в процедуру данных определяется величина функции *f*i(*Cmr*, *loc*)
2. Полученное значение сравнивается с радиусом поверхности разрушения с предыдущего шага (хранится в массиве *loc*). Если величина *f*i на текущем шаге превышает значение квадрата радиуса поверхности разрушения *r*i с предыдущего шага, то имеет место развитие поврежденности материала.

Параметр поврежденности от конкретного режима разрушения определяется формулой:



Общая поврежденность, влияющая на конкретный модуль, определяется матрицей влияния (формула (9)) на основании всех поврежденностей, полученных по различным режимам нагружения.

За отчетный период реализованы и протестированы алгоритмы для расчета поврежденности материала для шести режимов разрушения однонаправленного композиционного материала (поверхности разрушения *f*1-*f*6 согласно руководству КП1):

- режимы разрушения армирования (*f*1-*f*3): разрушение армирующего волокна при растяжении и сдвиге, разрушение при сжатии в направлении армирующего волокна и раздавливание композита в направлении перпендикулярном волокну

-разрешение связующего(*f*4-*f*6): сжатие в направлении плоскости слоя перпендикулярно направлению армирования (материальное направление 2), растяжение в направлении 2 и сдвиг в плоскостях 23 и 12, а также режим деламинации или расслоения (растяжение в направлении 3 и сдвиг в плоскостях 23 и 31).

## Результаты решения тестовых задач

Как видно из рисунка 100 в реализуемой модели каждый компонент тензора деформаций может влиять поврежденность сразу нескольких модулей материала. Для отладки алгоритмов разработаны и решены тестовые задачи, в которых механизмы разрушения разделены, а значит гораздо проще анализировать полученный результат и обнаруживать и устранять алгоритмические ошибки.

При решении задач использовались параметры модели однонаправленного КМ - UD S-2 Glass/SC15, приведенные в таблице 3.

Далее на графиках приводятся полученные результаты.

### Поверхность разрушения *f*1

Математическая формулировка указанной поверхности выражается уравнением (1). «Срабатывание» этого критерия возможно при наличии деформаций растяжения в направлении 1, либо деформаций сдвига в плоскостях 12 или 13.

Для отладки критерия решены следующие задачи:

Задача 1 - Растяжение материала в направлении 1 (направление армирования)

На рисунке 201 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 1 в условиях одноосного напряженного состояния. Горизонтальной черной линией отмечено предельное напряжение материала при растяжении в направлении 1 (параметр *X1T*). Видно, что максимальное напряжение в направлении армирования совпадает с этим значением. Дальше происходит деградация характеристик материала и напряжение падает. Причем за счет большого значения параметра скорости деградации (AM1 = 100), уменьшение напряжений происходит достаточно резко.

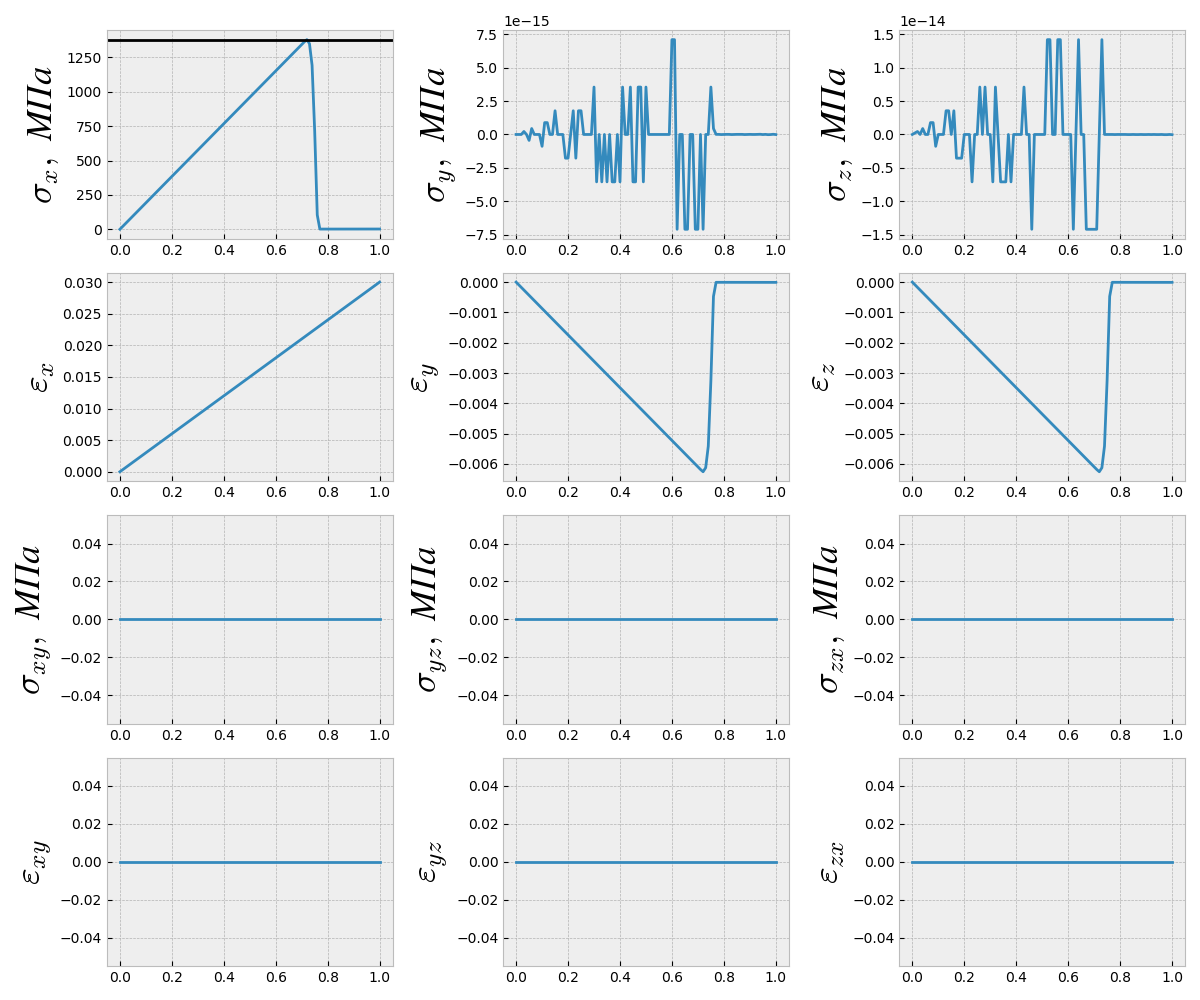


Рисунок 201 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 1 (одноосное напряжение)

На рисунке 202 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 1 в условиях одноосного деформированного состояния.

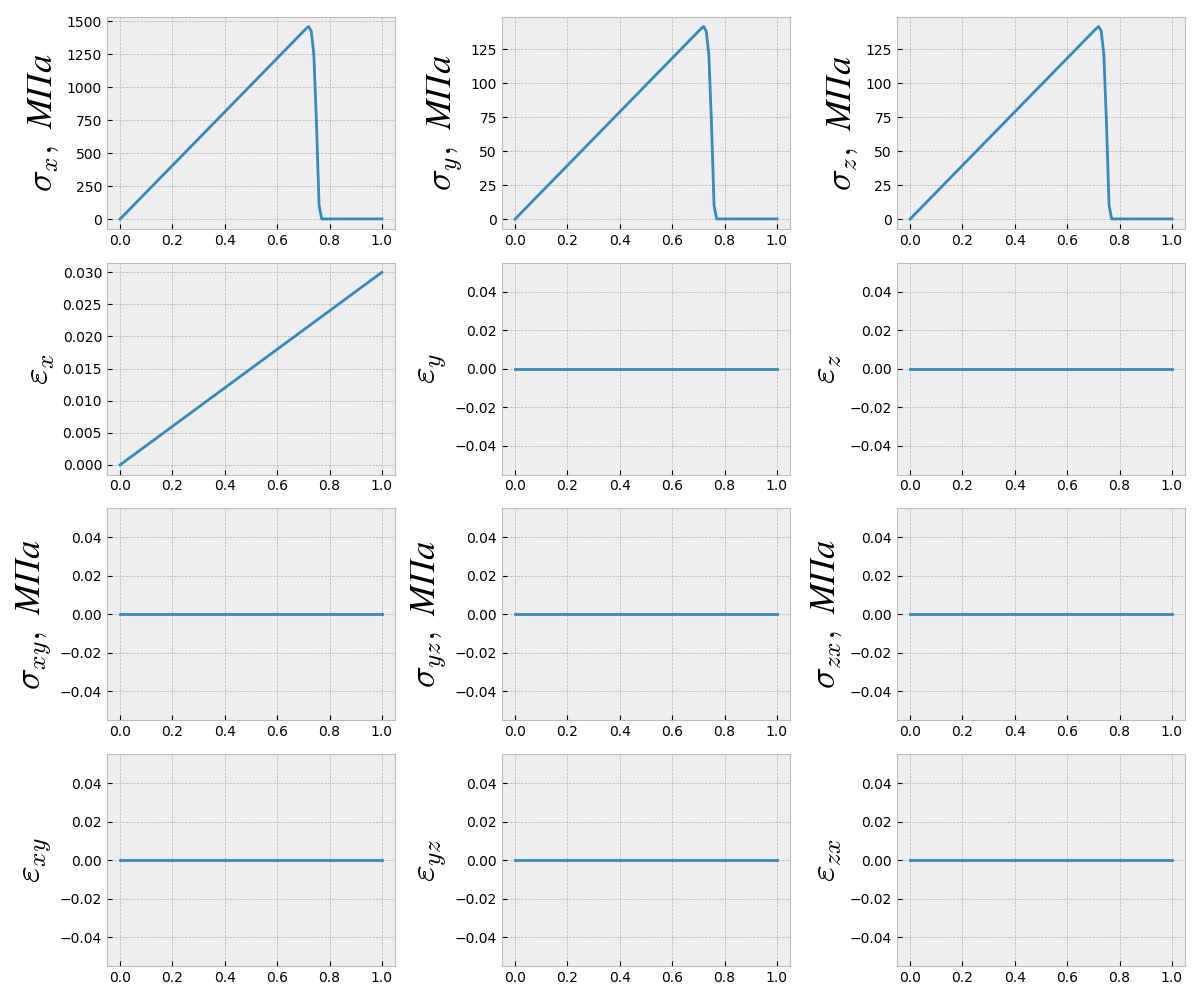


Рисунок 202 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 1 (одноосная деформация)

Задача 2 – Сдвиги в плоскости 12 и 31 (режим разрушения волокна)

На рисунках 203 и 4 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскостях 12 и 13. Горизонтальной черной линией отмечена прочность волокна при сдвиге (параметр *SFS*). Следует отметить, что для разделения эффектов разрушения волокна и матрицы в задачах 3 сдвиговая прочность связующего (параметры S12, S23 и S13) были искусственно завышены.

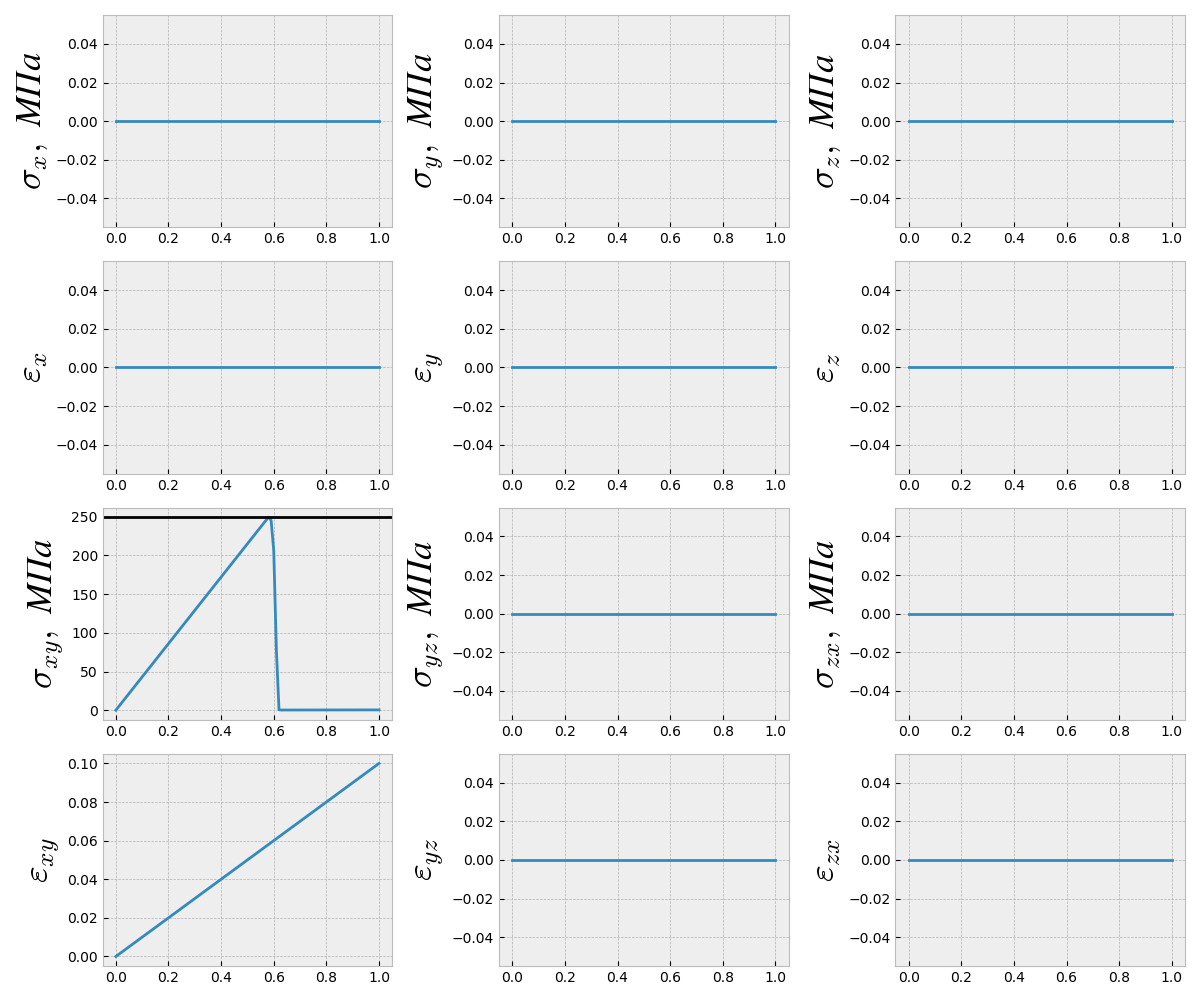


Рисунок 203 – Компоненты тензора напряжения (сверху) и деформации (снизу) для сдвига в плоскости 12 (режим разрушения волокна)

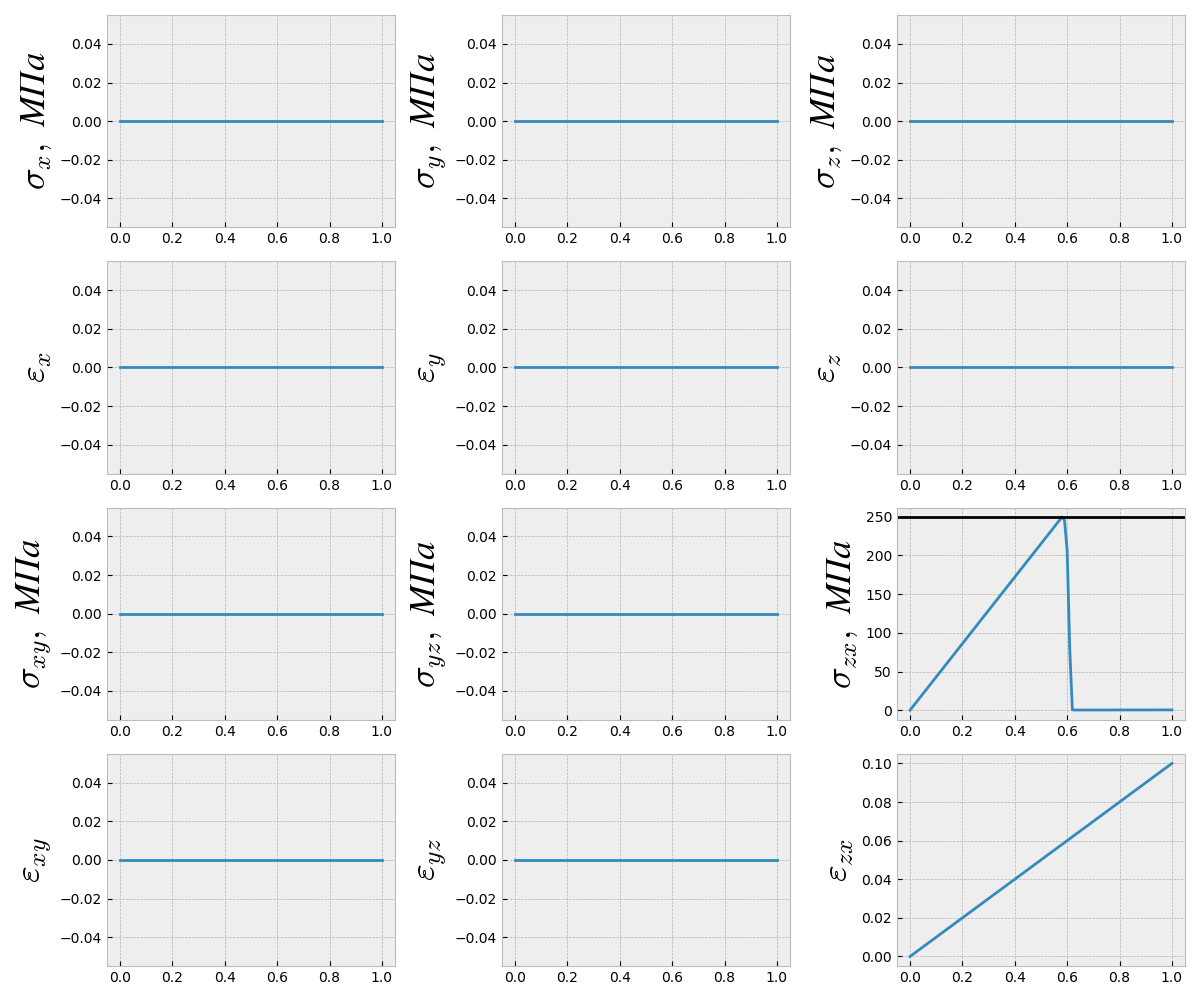


Рисунок 204 – Компоненты тензора напряжения (сверху) и деформации (снизу) для сдвига в плоскости 31 (режим разрушения волокна)

### Поверхность разрушения *f*2

Математическая формулировка указанной поверхности выражается уравнением (2). «Срабатывание» этого критерия возможно при наличии деформации сжатия в направлении 1.

Для отладки критерия решены следующие задачи:

Задача 3 – Сжатие в направлении 1

На рисунках 205 и 6 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия в направлении 1 в условиях одноосного напряжения и деформации соответственно. Горизонтальные черные линии на рисунке 205 отмечают исходную и остаточную прочности материала при сжатии. Более подробно алгоритм учета остаточной прочности материала описан ниже.

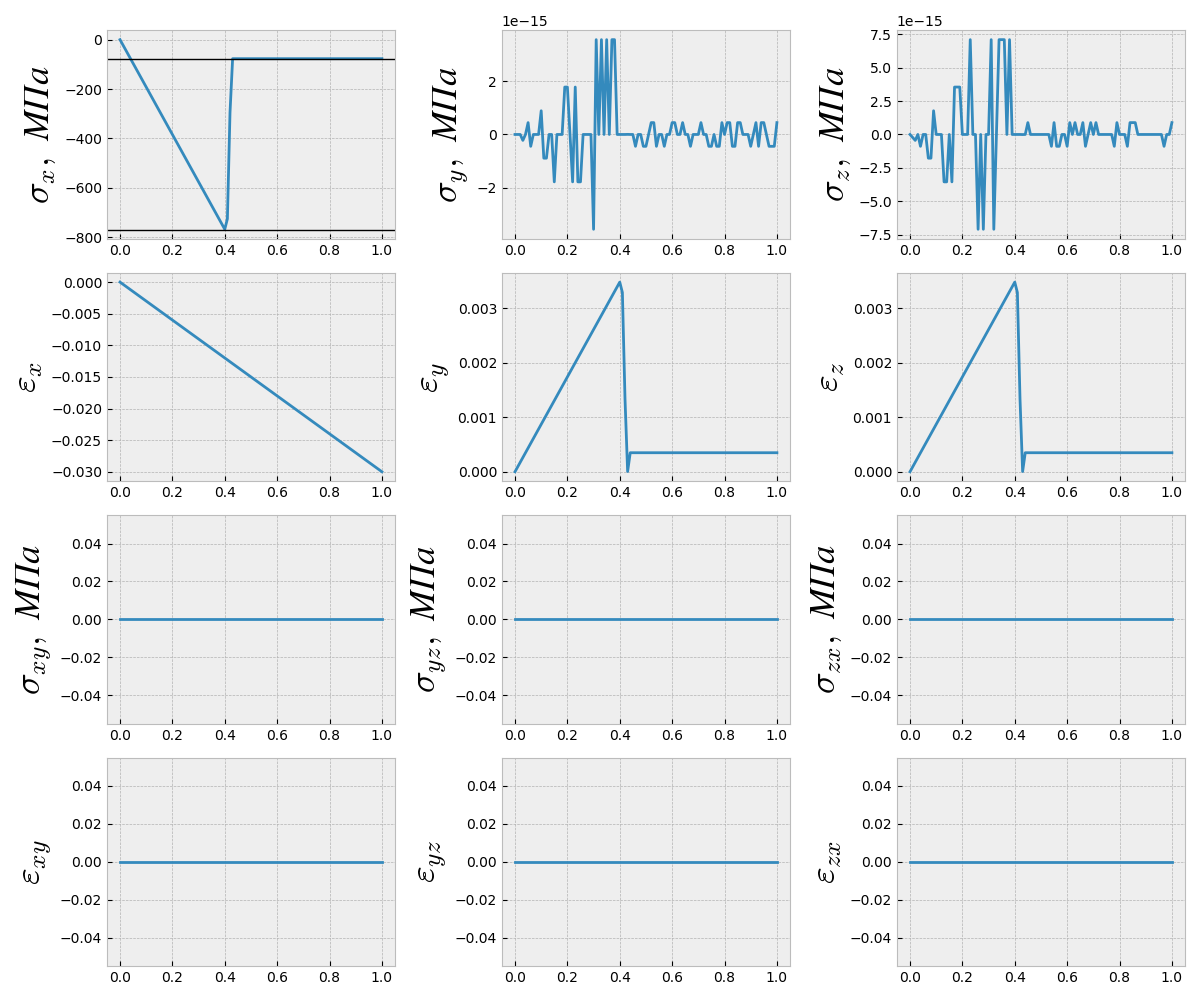


Рисунок 205 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия материала в направлении 1 (одноосное напряжение)

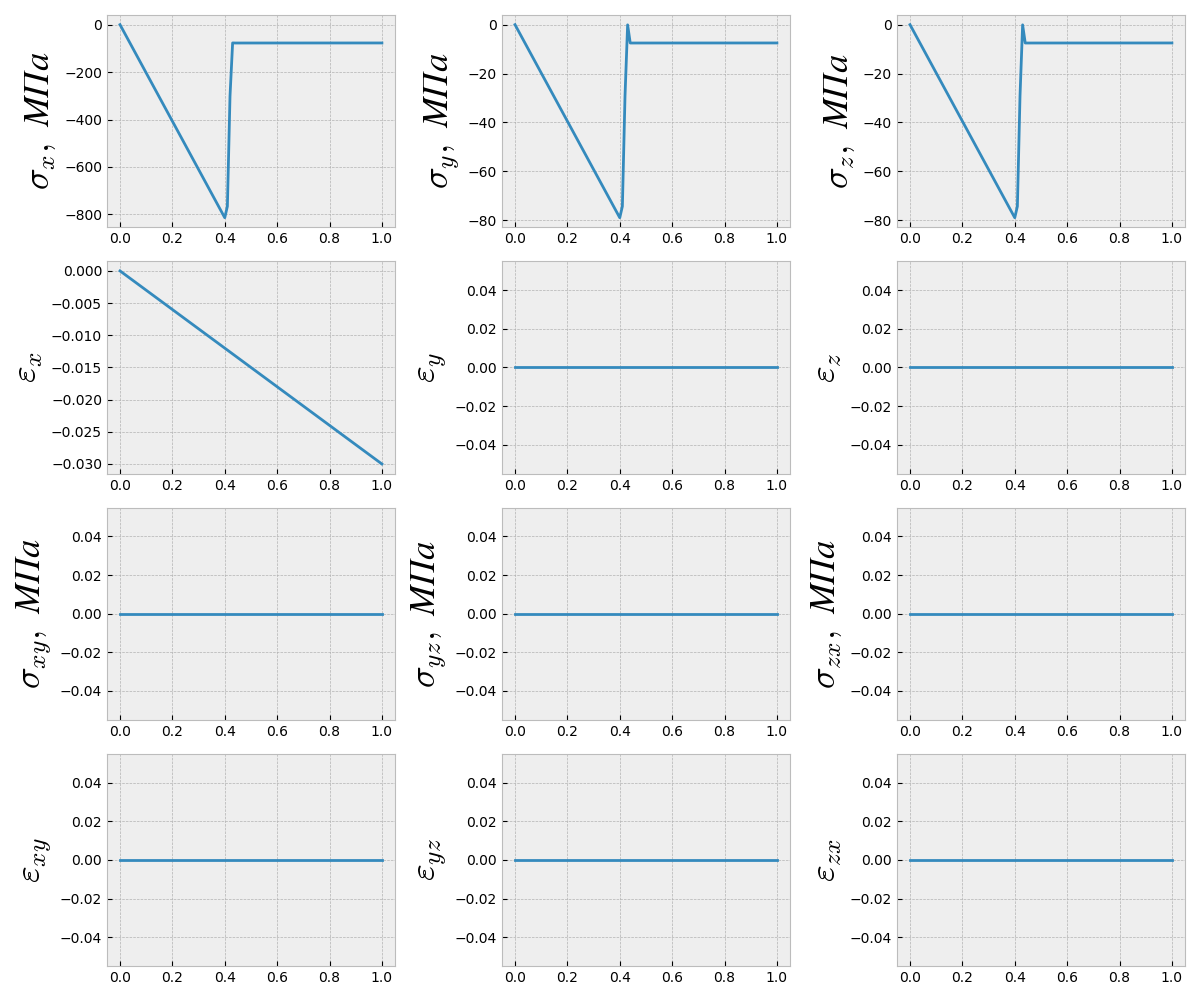


Рисунок 206 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия материала в направлении 1 (одноосная деформация)

### Поверхность разрушения *f*3

Математическая формулировка указанной поверхности выражается уравнением (3). «Срабатывание» этого критерия возможно при наличии деформации сжатия в направлении 3.

Для отладки критерия решены следующие задачи:

Задача 4 – Сжатие в направлении 3 (режим разрушения армирования)

На рисунках 207 и 8 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия в направлении 3 в условиях одноосного напряжения и деформации соответственно. Горизонтальная черная линия на рисунке 207 отмечает прочность материала при раздавливании (параметр *SFC*).

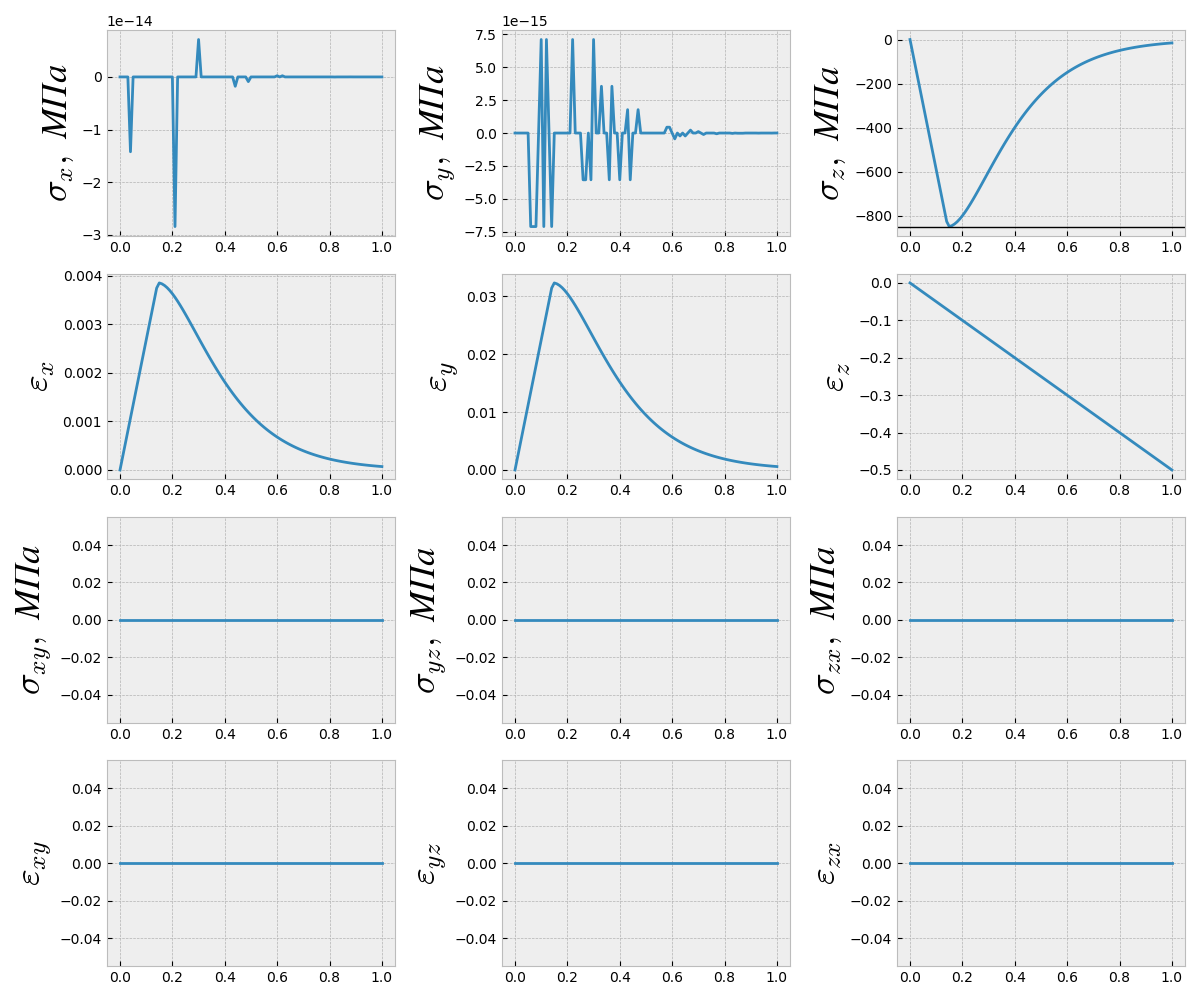


Рисунок 207 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия материала в направлении 3 (одноосное напряжение)

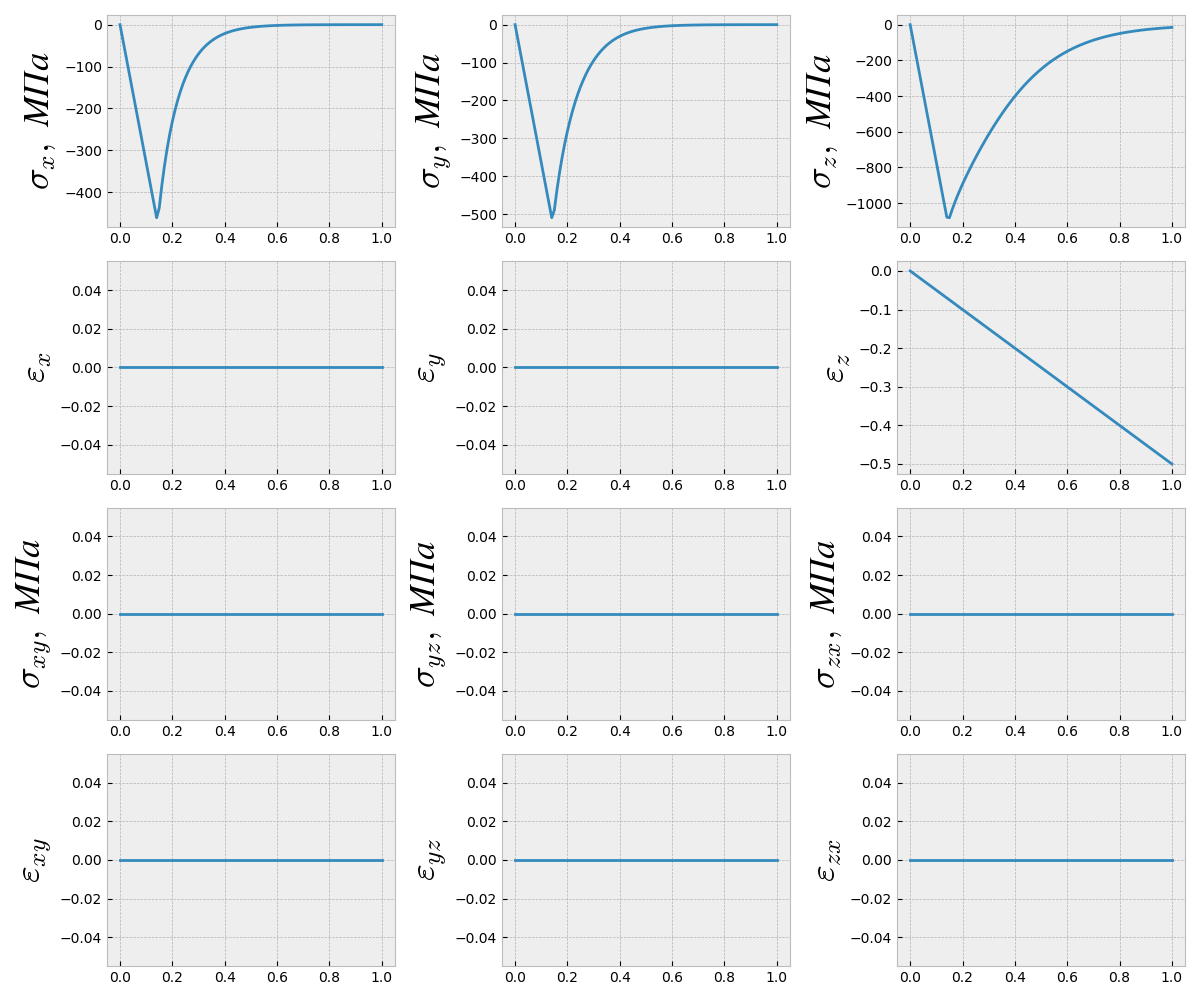


Рисунок 208 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия материала в направлении 3 (одноосная деформация)

### Поверхность разрушения *f*4

Математическая формулировка указанной поверхности выражается уравнением (4). «Срабатывание» этого критерия возможно при наличии деформации сжатия в направлении 2.

Для отладки критерия решены следующие задачи:

Задача 5 – Сжатие в направлении 2 (режим разрушения связующего)

На рисунках 209, 210 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия в направлении 2 в условиях одноосного напряжения и деформации соответственно. Горизонтальная черная линия на рисунке 209 отмечает прочность материала при сжатии в направлении 2 (параметр *X2C*).

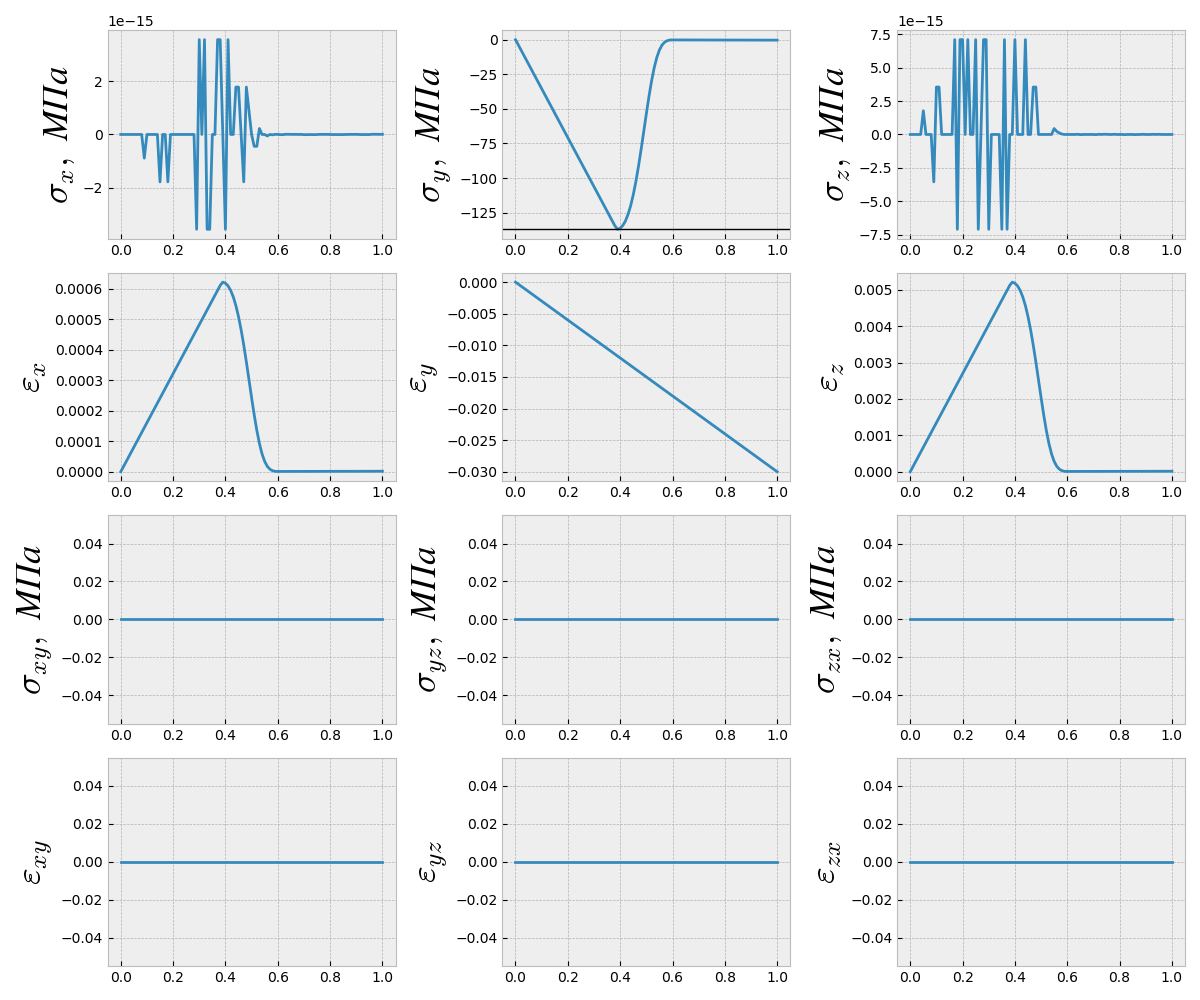


Рисунок 209 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия материала в направлении 2 (одноосное напряжение)

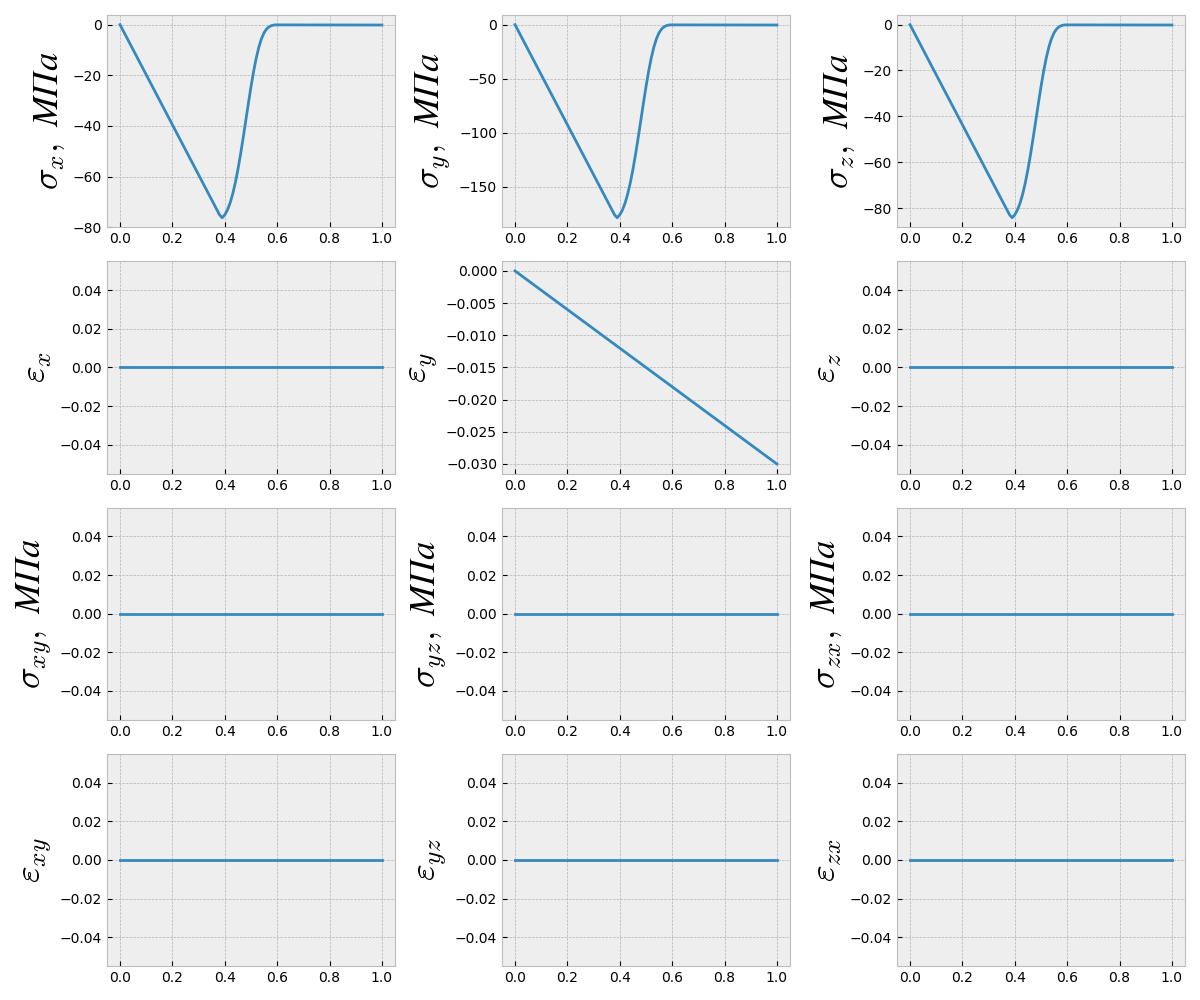


Рисунок 210 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сжатия материала в направлении 2 (одноосная деформация)

### Поверхность разрушения *f*5

Математическая формулировка указанной поверхности выражается уравнением (5) и (7). «Срабатывание» этого критерия возможно при наличии деформации растяжения в направлении 2, а также сдвиговых деформаций в плоскостях 23 и 12. Причем наличие сжимающей деформации в направлении 2 влияет на напряжение «срабатывания» сдвиговых критериев прочности.

Для отладки критерия решены следующие задачи:

Задача 6 – растяжение в направлении 2

На рисунках 211, 212 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения в направлении 2 в условиях одноосного напряжения и деформации соответственно. Горизонтальная черная линия на рисунке 211 отмечает прочность материала при растяжении в направлении 2 (параметр *X2T*).

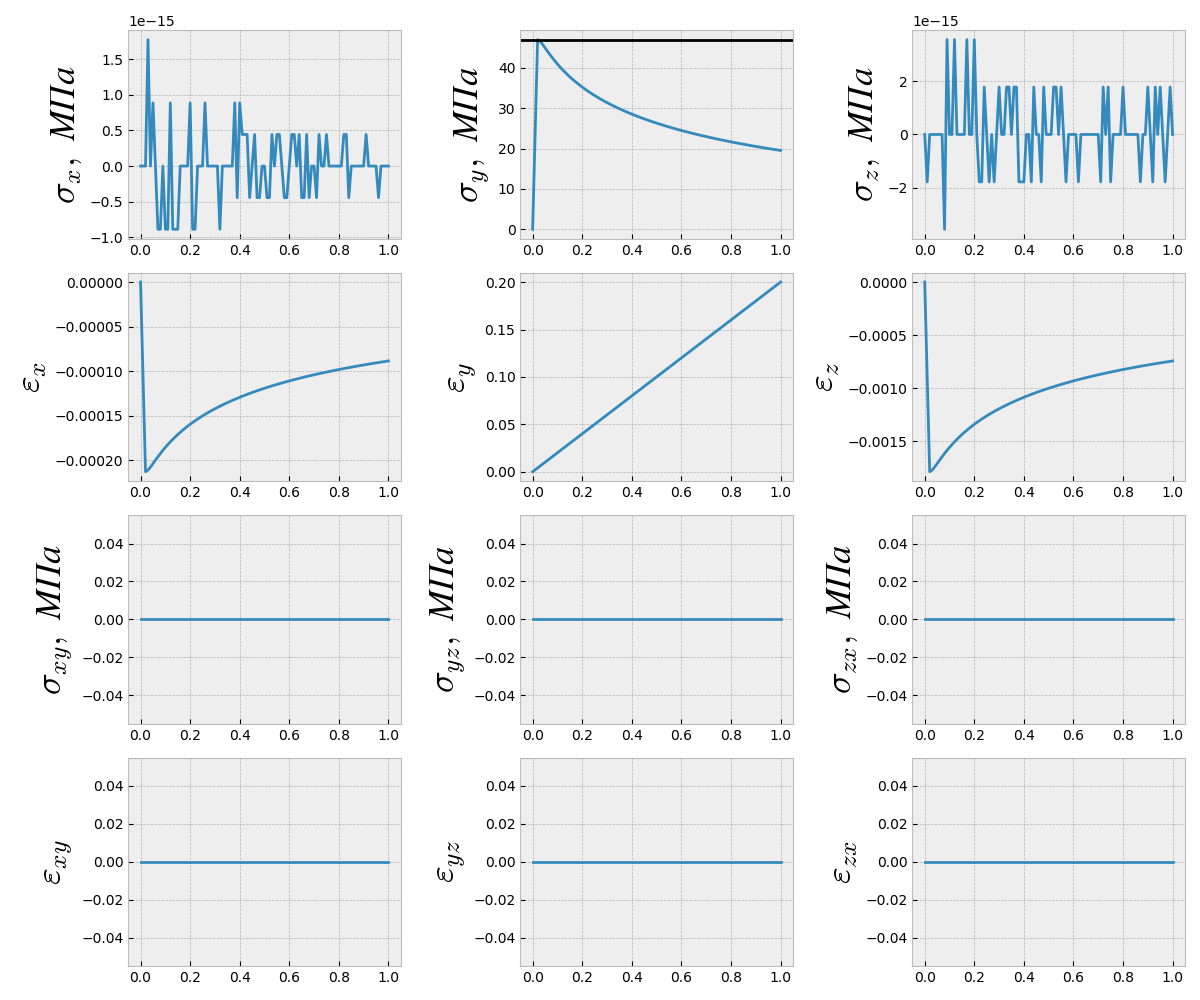


Рисунок 211 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 2 (одноосное напряжение)

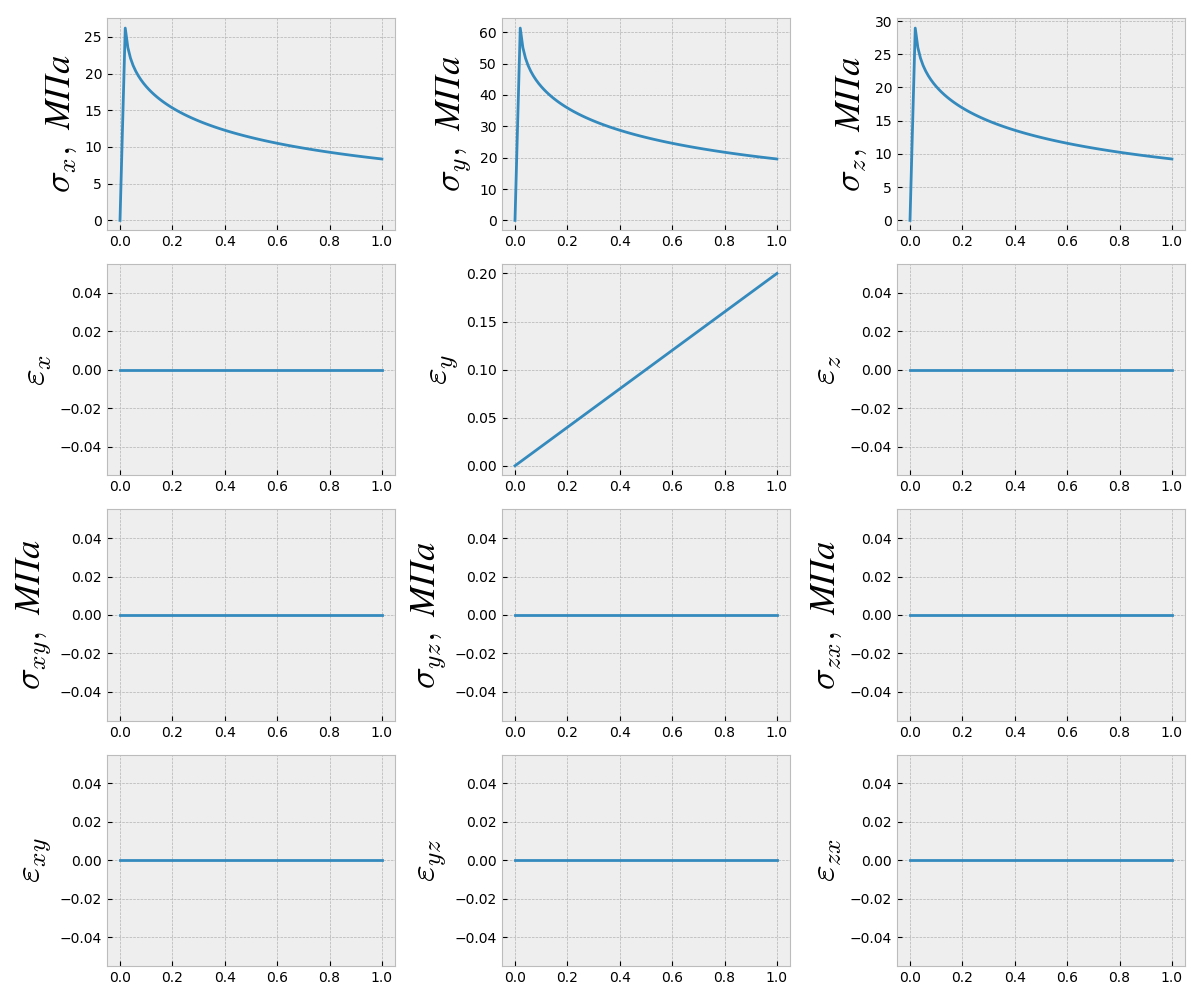


Рисунок 212 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 2 (одноосная деформация)

Задача 7 – Простые сдвиги в плоскостях 12 и 23 (режим разрушения связующего)

На рисунках 213, 214 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскостях 12 и 23. Горизонтальными черными линиями отмечены прочности связующего при сдвигах в плоскостях 12 и 23 (параметры *S12* и *S23* соответственно). Следует отметить, что для разделения эффектов разрушения волокна и матрицы в задачах 7 сдвиговая прочность армирования (параметры SFS) были искусственно завышены. Также в задаче сдвига в плоскости 23 коэффициент масштабирования деламинации *SDELM* был задан равным 1, для исключения более раннего срабатывания поверхности разрушения *f*6.

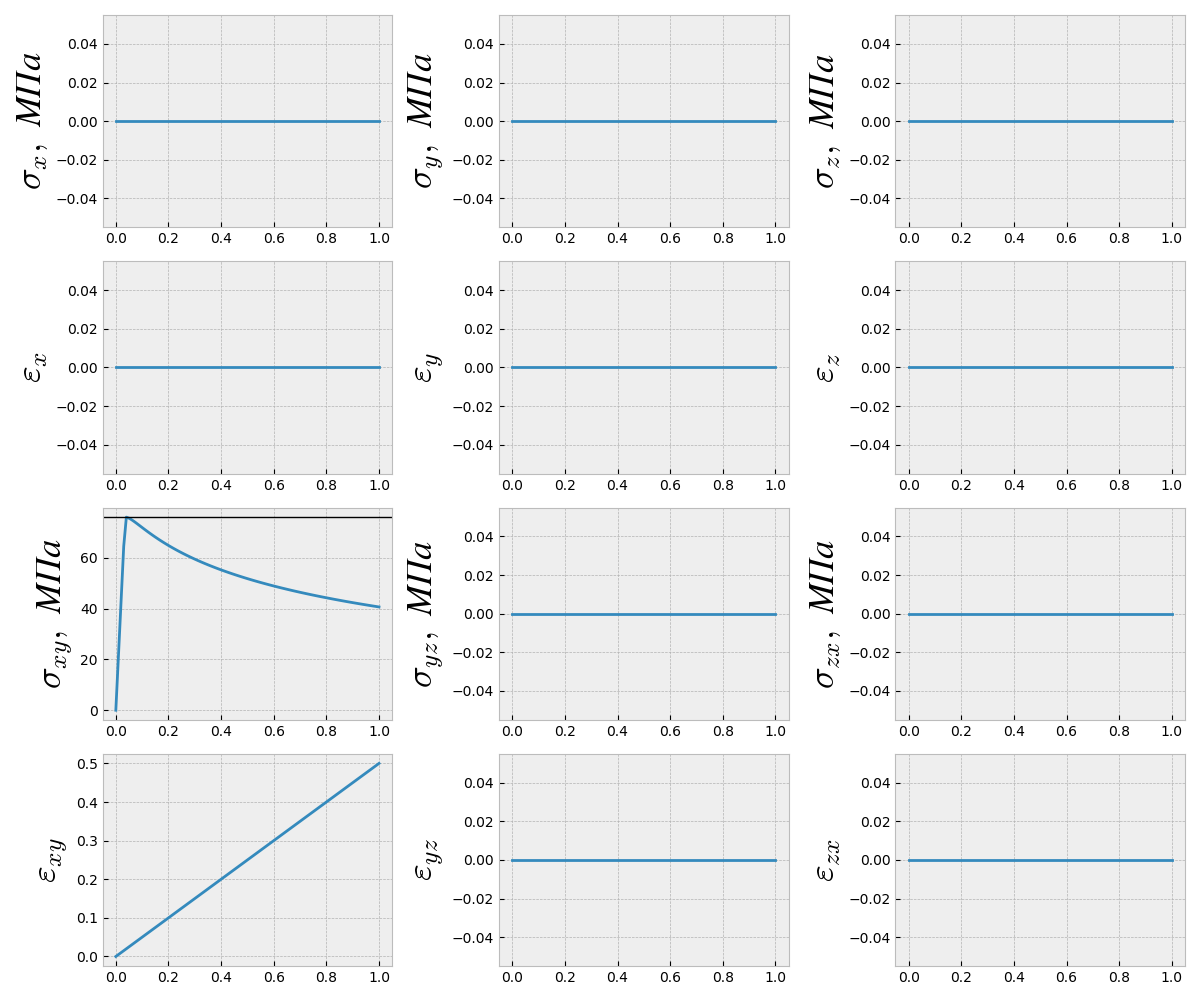


Рисунок 213 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 12 (режим разрушения связующего)

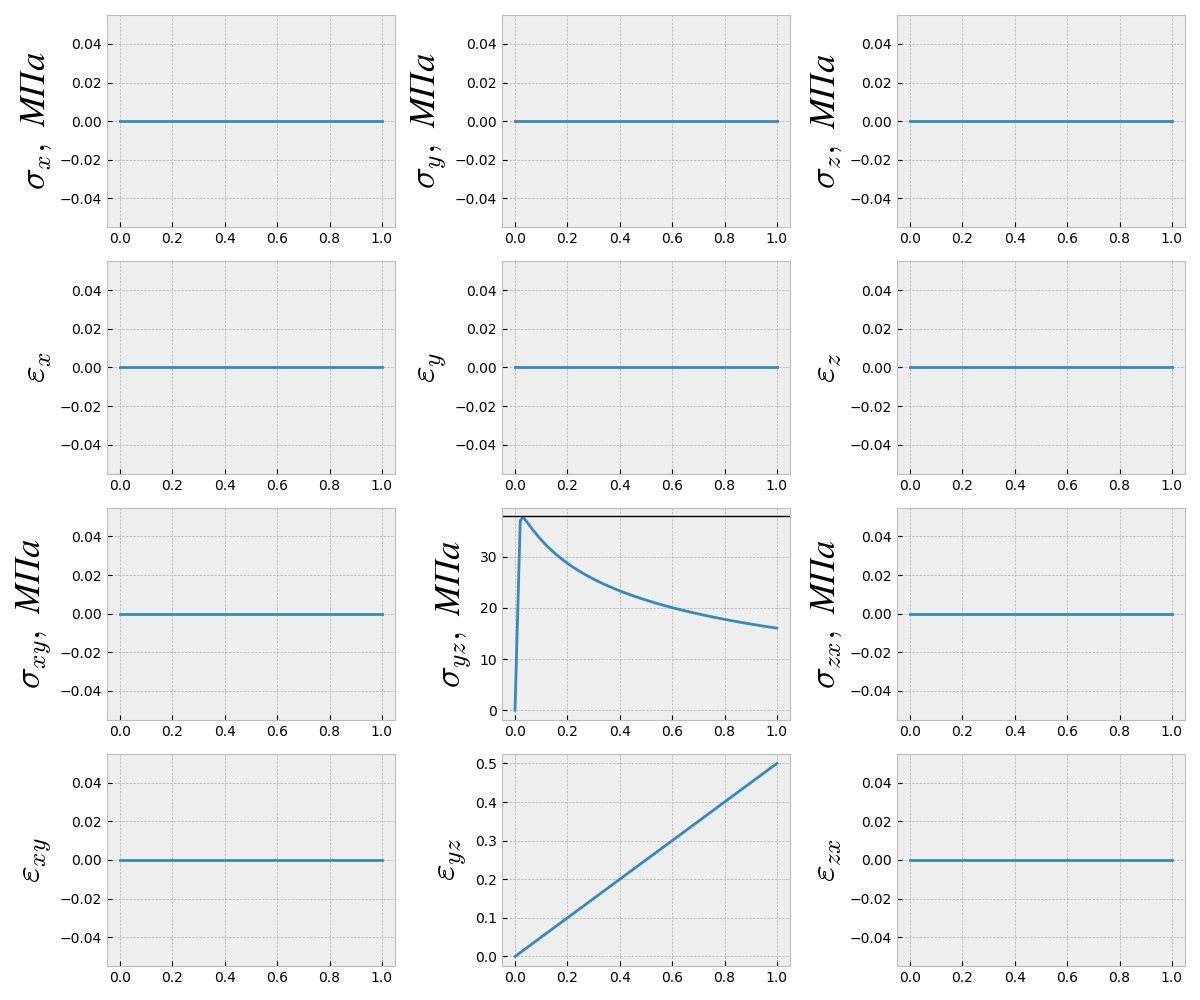


Рисунок 214 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 23 (режим разрушения связующего)

Задача 8 – Сдвиги в плоскостях 12 и 23 (режим разрушения связующего) при наличии сжимающей деформации в направлении 2

На рисунках 215, 216 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскостях 12 и 23 при наличии сжимающей деформации в направлении 2. Горизонтальными черными линиями отмечены прочности связующего при сдвигах в плоскостях 12 и 23 (параметры *S12* и *S23* соответственно). Следует отметить, что для разделения эффектов разрушения волокна и матрицы в задачах 8 сдвиговая прочность армирования (параметры SFS) были искусственно завышены. Также в задаче сдвига в плоскости 23 коэффициент масштабирования деламинации *SDELM* был задан равным 0, для исключения более раннего срабатывания поверхности разрушения *f*6. Видно, что наличие сжимающего напряжения в направлении 2 приводит к увеличению напряжения, при котором начинается деградация свойств.

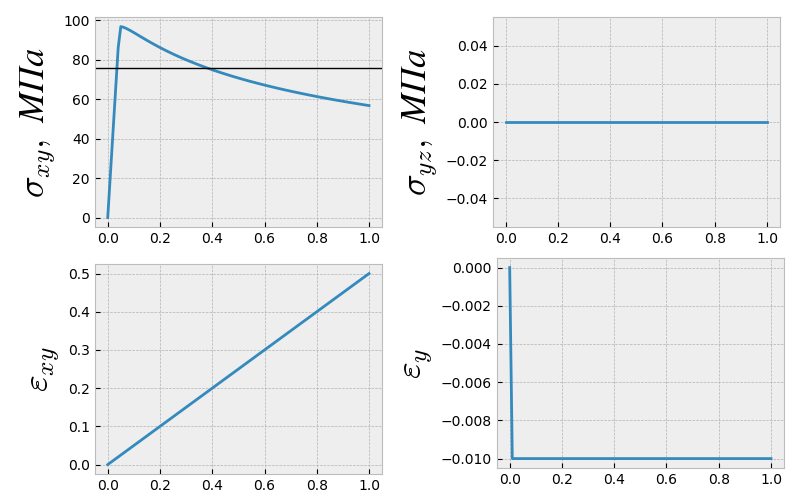


Рисунок 215 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 12 (режим разрушения связующего) при наличии деформации сжатия в направлении 2

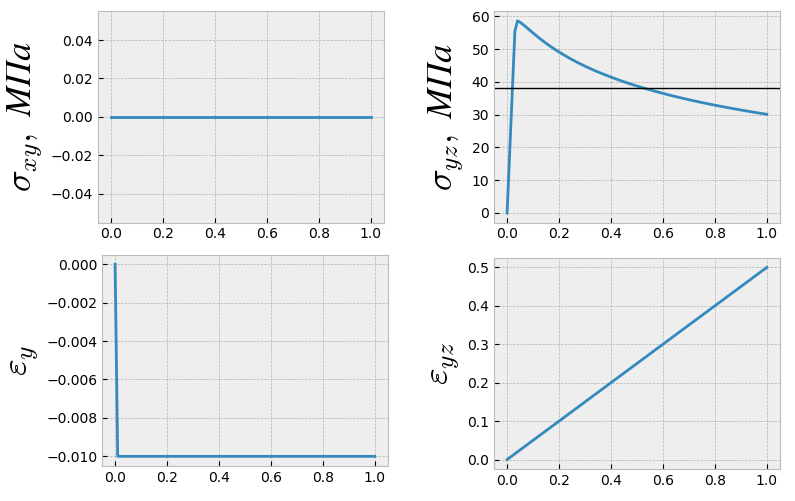


Рисунок 216 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 23 (режим разрушения связующего) при наличии деформации сжатия в направлении 2

### Поверхность разрушения *f*6

Математическая формулировка указанной поверхности выражается уравнением (6), (7). «Срабатывание» этого критерия возможно при наличии деформации растяжения в направлении 3, а также сдвиговых деформаций в плоскостях 23 и 13. Причем наличие сжимающей деформации в направлении 3 влияет на напряжение «срабатывания» сдвиговых критериев прочности.

Для отладки критерия решены следующие задачи:

Задача 9 – растяжение в направлении 3

На рисунках 217, 218 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения в направлении 3 в условиях одноосного напряжения и деформации соответственно. Горизонтальная черная линия на рисунке 217 отмечает прочность материала при растяжении в направлении 3 с учетом коэффициента масштабирования (параметр *X3T/SDELM*).

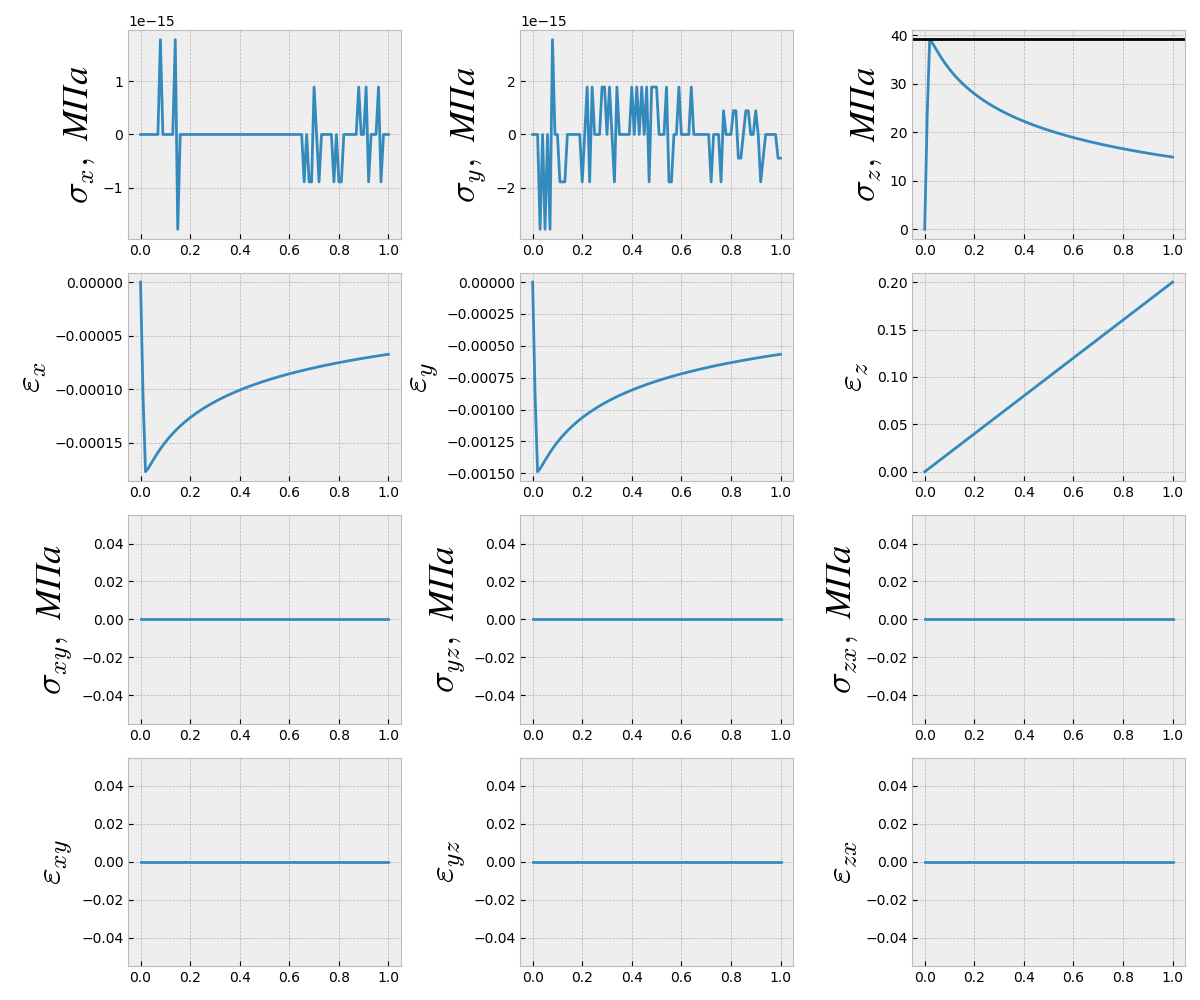


Рисунок 217 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 3 (одноосное напряжение)

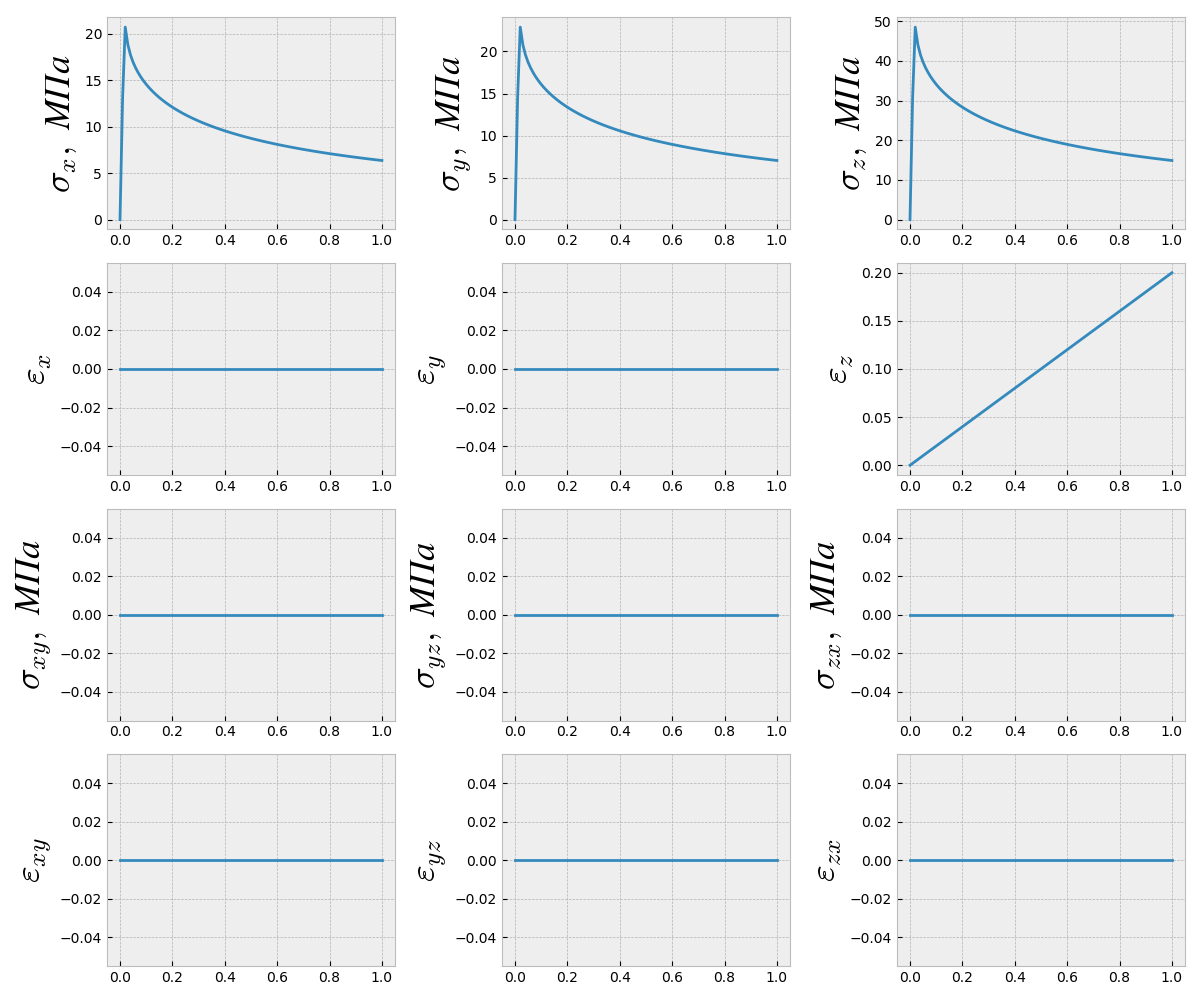


Рисунок 218 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи растяжения материала в направлении 3 (одноосная деформация)

Задача 10 – Простые сдвиги в плоскостях 13 и 23 (режим разрушения связующего)

На рисунках 219, 220 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскостях 13 и 23. Горизонтальными черными линиями отмечены прочности связующего при сдвигах в плоскостях 13 и 23 с учетом коэффициента масштабирования (параметры *S12/SDELM* и *S23/SDELM* соответственно). В задаче сдвига в плоскости 13 для исключения срабатывания критерия по разрушению волокна *f*1 значение сдвиговой прочности волокна *SFS* было искусственно завышено.

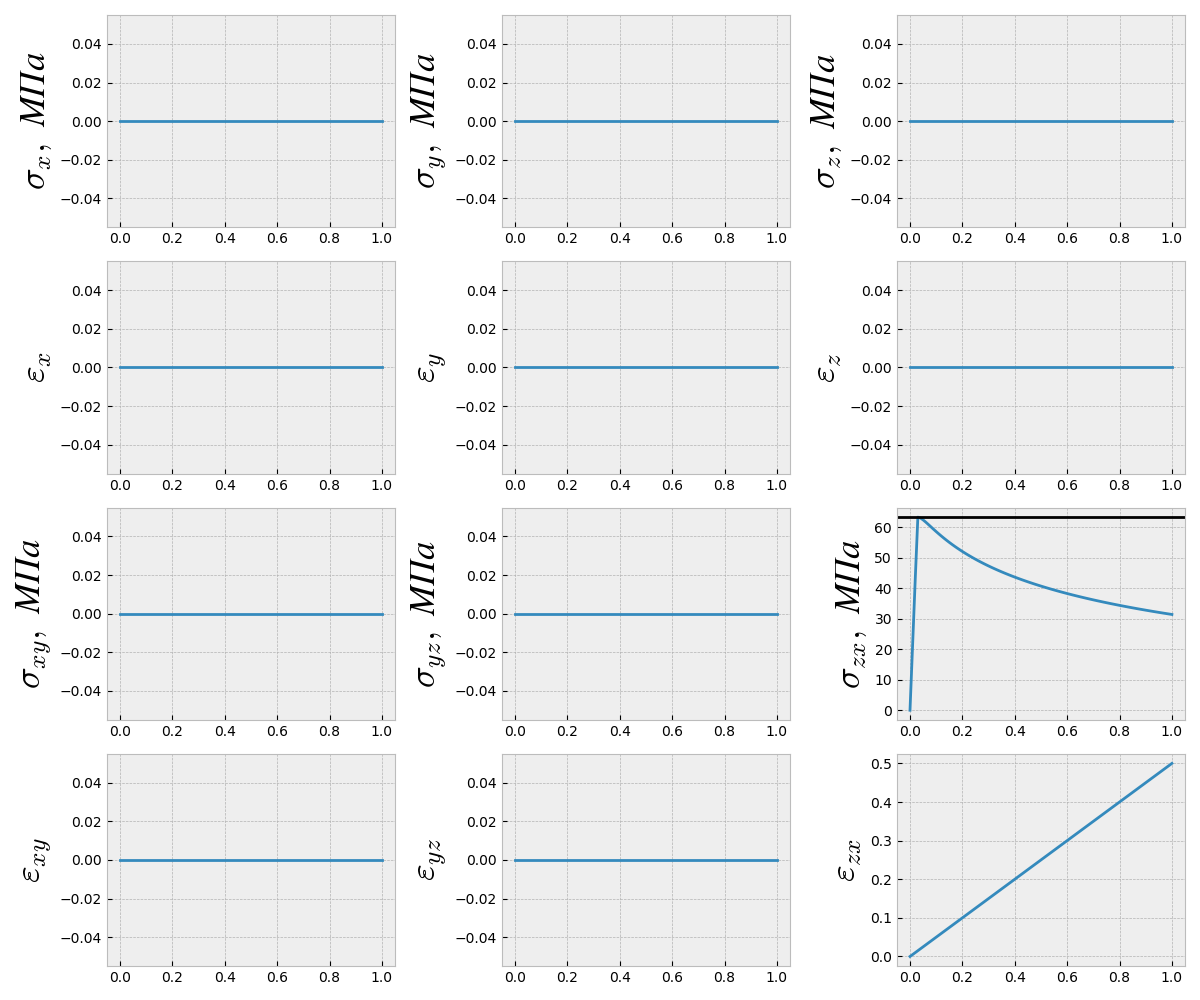


Рисунок 219 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 13 (режим разрушения связующего)

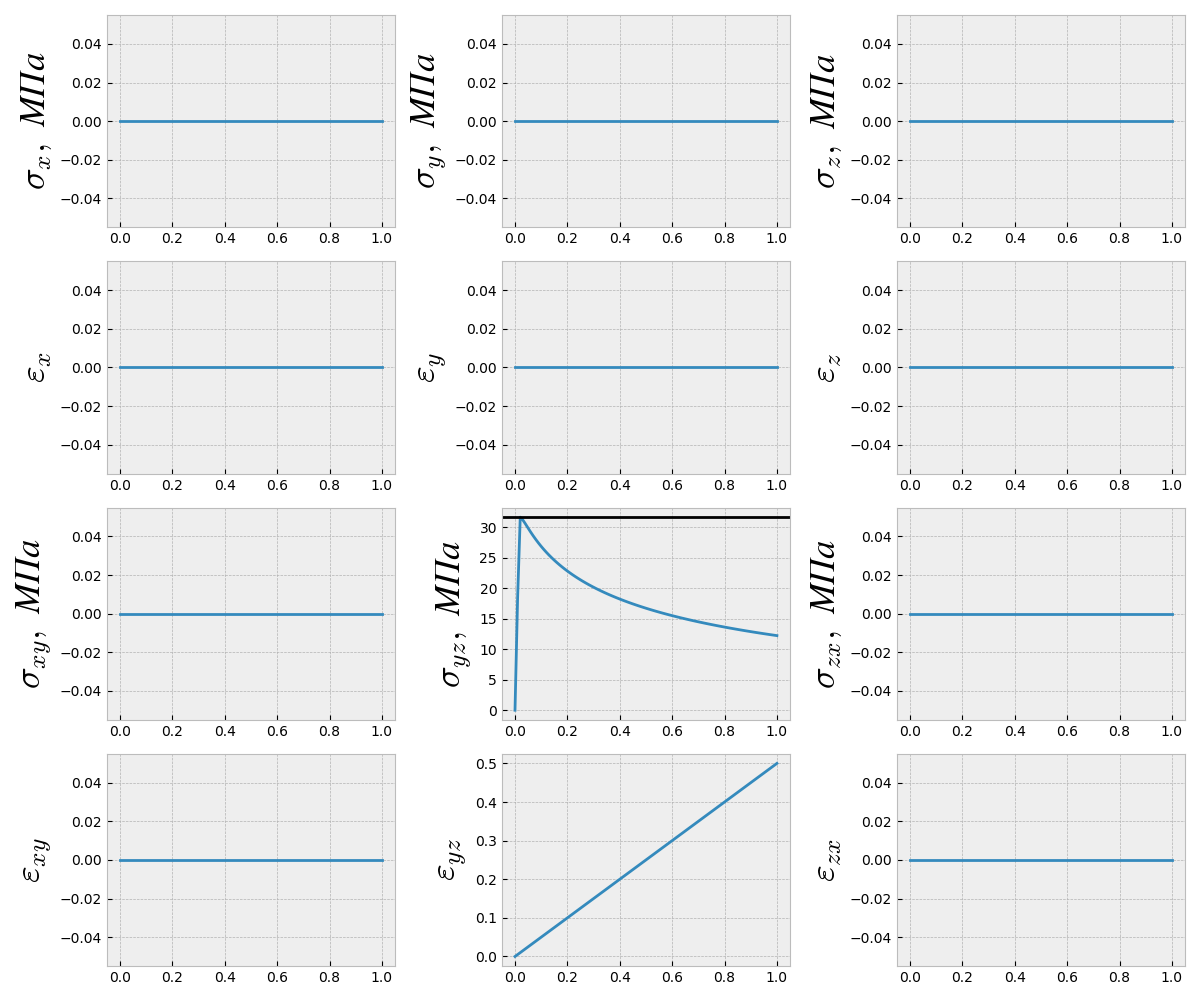


Рисунок 220 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 23 (режим разрушения связующего с учетом деламинации)

Задача 11 – Сдвиги в плоскостях 13 и 23 (режим разрушения связующего) при наличии сжимающей деформации в направлении 3

На рисунках 221, 222 приводятся компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскостях 13 и 23 при наличии сжимающей деформации в направлении 3. Горизонтальными черными линиями отмечены прочности связующего при сдвигах в плоскостях 13 и 23 с учетом масштабирующего коэффициента (параметры *S13/SDELM* и *S23/SDELM* соответственно). Следует отметить, что для разделения эффектов разрушения волокна и матрицы в задачах 11 сдвиговая прочность армирования (параметры *SFS*) были искусственно завышены. Видно, что наличие сжимающего напряжения в направлении 3 приводит к увеличению напряжения, при котором начинается деградация свойств.

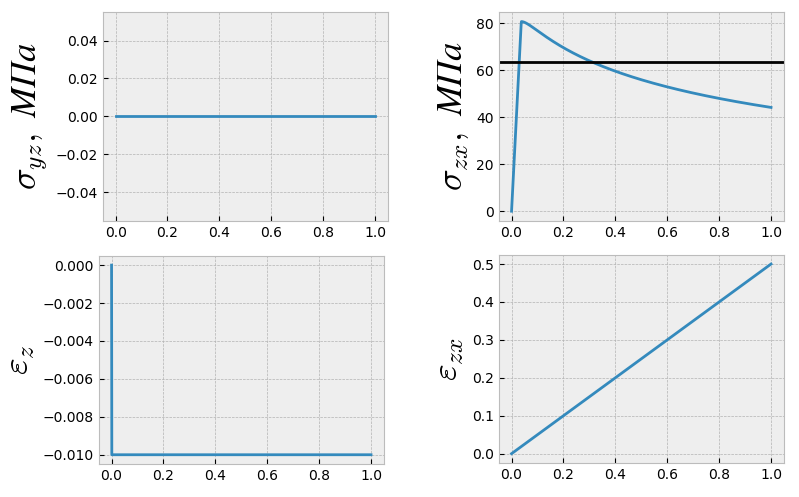


Рисунок 221 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 13 (режим разрушения связующего) при наличии деформации сжатия в направлении 3

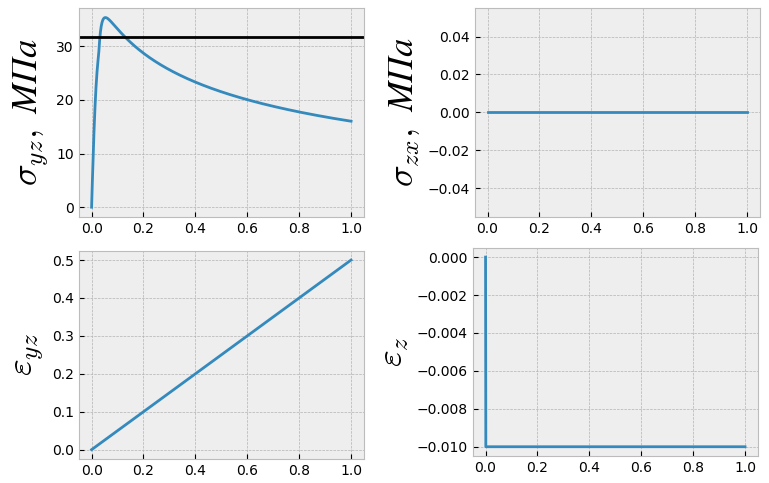


Рисунок 222 – Компоненты тензора напряжения (сверху) и деформации (снизу) для задачи сдвига в плоскости 23 (режим разрушения связующего) при наличии деформации сжатия в направлении 3

### Алгоритм учета остаточной прочности материала при сжатии в направлении армирования

Разработано и протестировано несколько вариантов учета остаточной прочности композита при сжатии. Выбрана модель остаточной прочности, в которой разгрузка и повторное нагружение материала при достижении остаточной прочности происходит по линии, отвечающей модулю поврежденного материала, при котором произошло снижение прочности до остаточного значения. Критерием достижения остаточной прочности является пересечение уровня остаточной прочности ниспадающей ветвью кривой деформирования при выполнении условия приращения радиуса соответствующей поверхности разрушения (т.е. переход через линию остаточной прочности при росте поврежденности материала).

Используемы алгоритм иллюстрируется на рисунках 223, 224. Задана история изменения осевой деформации по закону, изображенному в правой части рисунка 222. Вначале деформация сжатия увеличивается до 1.5 %, при этом достигается критическая точка 1 и начинается накопление поврежденности до точки 2. В точке 2 деформация снимается до 0, происходит разгрузка. Напряжение также снижается до 0. Затем проводится повторное нагружение до уровня сжимающей деформации 3%. При этом материал переходит через точку 5 начала приращения параметра поврежденности и в точке 6 достигается уровень остаточной прочности, при этом радиус поверхности разрушения продолжает расти. Срабатывает механизм остаточной прочности и с ростом деформации уровень напряжения сохраняется на значении остаточной прочности. Долее происходит разгрузка материала и повторное нагружение. Поведение материала при достижении остаточной прочности напоминает идеально-упруго пластическую модель. Модуль упругой ветви материала, достигшего остаточной прочности, хранится в массиве *loc*.

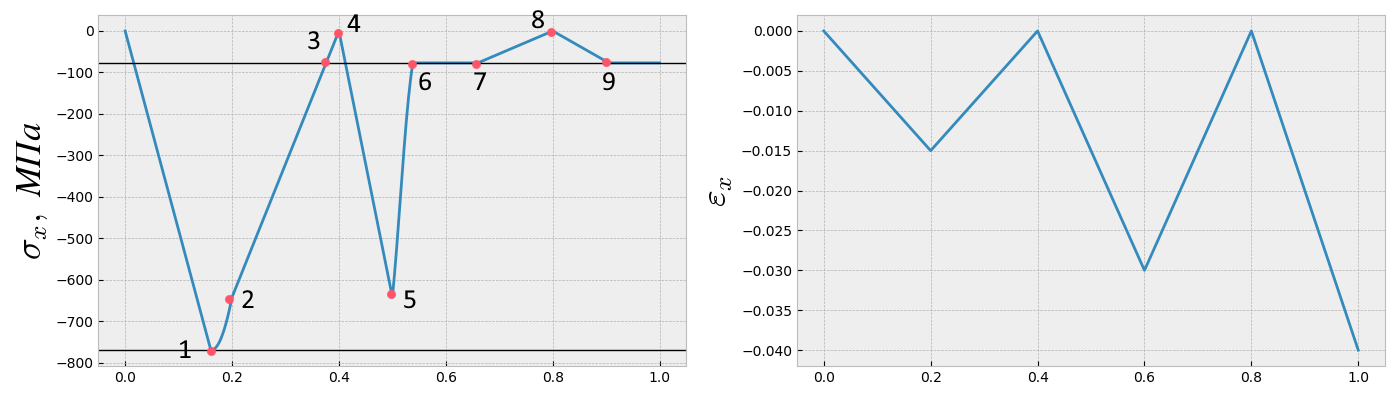


Рисунок 223 – Иллюстрация алгоритма учета остаточной прочности

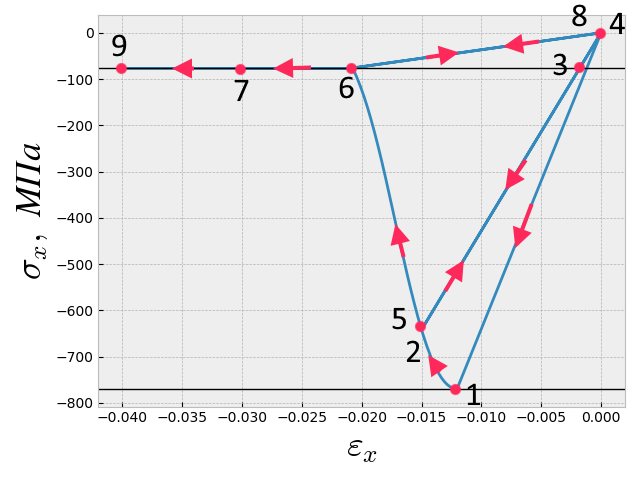


Рисунок 224 – Иллюстрация алгоритма учета остаточной прочности

### Трение разрушенных поверхностей

Отработан алгоритм учета взаимодействия поверхностей, образовавшихся при расслоении материала, посредством трения при наличии соответствующих сжимающих напряжений. Критерием появления сдвиговых напряжений, обусловленных трением, является наличие сжимающих напряжений в направлении, перпендикулярном плоскости сдвига, и наличие ненулевой скорости сдвиговой деформации в плоскости. Рисунок 225 иллюстрирует эффект внутреннего трения в задаче сдвига в плоскости 13 при наличии сжимающего напряжения в направлении 3.

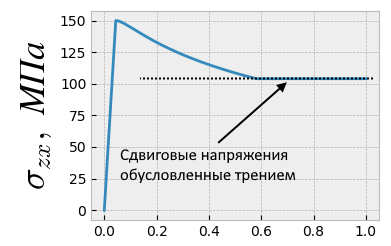


Рисунок 225 – Сдвиговые напряжения, обусловленные трением в задаче сдвига в плоскости 13 при наличии сжимающего напряжения в направлении 3

Все реализованные программные алгоритмы перенесены в процедуру, реализующую пользовательскую модель материала в ПП ЛОГОС. Подготовлены входные файлы для ПП ЛОГОС для дальнейшей отладки и проверки процедур.

## Заключение по разделу 1

В рамках внедрения в ПП ЛОГОС модели композиционного материала с учетом прогрессирующего разрушения, сложного напряженного состояния и эффектов скорости деформации (аналог MAT\_COMPOSITE\_MSC\_DMG) решены следующие задачи:

1. На основе информации о моделях MAT\_161/162 (MAT\_COMPOSITE\_MSC) КПО по основным источникам сформулирована математическая модель прогрессирующего разрушения однонаправленного композиционного материала. Проанализированы основные особенности реализуемой модели.
2. Сформулированы требования к входным параметрам модели.
3. Для тестирования и отладки создан программный макет модели на языке *python*.
4. В рамках макета реализован полный набор критериев разрушения, накопления повреждений и деградации механических характеристик для однонаправленного композиционного материала: три для режимов разрушения армирующего волокна (поверхности *f1*‑*f3*) и три – для режимов разрушения матрицы (поверхности *f4*‑*f6*).
5. Разработано и протестировано несколько вариантов учета остаточной прочности композита при сжатии в направлении волокна. Выбрана модель остаточной прочности, в которой разгрузка и повторное нагружение материала при достижении остаточной прочности происходит по линии, отвечающей модулю поврежденного материала, при котором произошло снижение прочности до остаточного значения. Критерием достижения остаточной прочности является пересечение уровня остаточной прочности ниспадающей ветвью кривой деформирования при выполнении условия приращения радиуса соответствующей поверхности разрушения (т.е. переход через линию остаточной прочности при росте поврежденности материала).
6. Отработан алгоритм учета взаимодействия поверхностей, образовавшихся при расслоении материала, посредством трения при наличии соответствующих сжимающих напряжений. Критерием появления сдвиговых напряжений, обусловленных трением, является наличие сжимающих напряжений в направлении, перпендикулярном плоскости сдвига, и наличие ненулевой скорости сдвиговой деформации в плоскости.
7. С использованием макета сформулированы и решены тестовые задачи, для анализа работы отдельных механизмов разрушения. Получены зависимости напряжений от времени и деформации при простых монотонных нагружениях на элементарном объеме (при заданных законах изменения компонент тензора деформации). Анализ полученных результатов позволил выявить и исправить ряд логических ошибок, допущенных в программной реализации математических моделей, а также провести оптимизацию расчетных алгоритмов. Данные, полученные на исправленном макете, качественно и количественно соответствуют ожидаемым результатам.
8. Отлаженные на макете алгоритмы перенесены в подпрограмму, реализующую пользовательскую модель материала ПП ЛОГОС.

Таким образом, все работы по направлению «Разработка и реализация для объемных конечных элементов модели композиционного материала с учетом прогрессирующего разрушения и динамического упрочнения» выполнены в полном объеме.

<stop><\stop>

## Список источников к разделу 1

1. A PROGRESSIVE COMPOSITE DAMAGE MODEL FOR UNIDIRECTIONAL AND WOVEN FABRIC COMPOSITES. UD-CCM Updateson MAT162 USER MANUAL. Version 15A-2015 // Materials Sciences Corporation (MSC) &University of Delaware Center for Composite Materials (UD-CCM). April 2015 /Technical Support . Bazle Z. (Gama) Haque.46 p.

2. Yen, C.F., (2002), “Ballistic Impact Modeling of Composite Materials,” Proceedings of 7th International LS-DYNA Users Conference, May, 2002, Dearborn, Michigan, pp.6.15-6.26.

3. Matzenmiller, A., Lubliner, J., and Taylor, R.L. (1995). “A Constitutive Model for Anisotropic Damage in Fiber-Composites,” Mechanics of Materials, 20, pp. 125-152.

4. Xiao, J. R., Gama, B. A., and Gillespie Jr., J. W., “Progressive Damage and Delamination in Plain Weave S-2 Glass/SC-15 Composites under Quasi-Static Punch Shear Loading,” Composite Structures, 2007, Vol. 78, pp. 182-196.

5. Gama, B. A., Travis A. Bogetti, and Gillespie, J. W. (1009). “Progressive Damage Modeling of Plain-Weave Composites using LS-Dyna Composite Damage Model MAT162,” Proceedings and CD Rom of 7th European LS-DYNA Conference, May 14-15, Salzburg, Austria, 2009.

6. Gama, B. A., and Gillespie, J. W. “Finite Element Modeling of Impact, Damage and Penetration of Thick-Section Composites,” International Journal of Impact Engineering, 2011, Vol. 38, pp. 181-197.

7. Impact Response of Laminated and Sandwich Composites by U. K. Vaidya // Impact Engineering of Composite Structures. Editors: Abrate, Serge. 2011, рр. 97-192.

48

8. B. Z. (Gama) Haque and J. W. Gillespie Jr.PENETRATION AND PERFORATION OF COMPOSITE STRUCTURES // Mechanical Engineering Research Journal. , 2013. Vol. 9, pp. 37-42.

9. Jordan, J. B., Naito, C. J., and (Gama) Haque, B. Z. Progressive damage modeling of plain weave E-glass/phenolic composites. Composites B, Vol. 61, May 2014, pp. 315-323.

Приложение 1 – Текст программы – текущая реализация пользовательского материала в ПП ЛОГОС

**module** mat\_msc\_functions

**implicit** none

**real**, **parameter** :: PI=3.141592653589793

contains

**function** inverse2(m) result(N)

**real**, **dimension**(3,3), intent(in):: m

**real**, **dimension**(3,3):: N

**real**:: a11, a12, a13, a23, a22, a33, d

a11 = m(3,3)\*m(2,2)-m(2,3)\*m(2,3)

a12 = m(1,3)\*m(2,3)-m(3,3)\*m(1,2)

a13 = m(1,2)\*m(2,3)-m(1,3)\*m(2,2)

a22 = m(3,3)\*m(1,1)-m(1,3)\*m(1,3)

a23 = m(1,2)\*m(1,3)-m(1,1)\*m(2,3)

a33 = m(1,1)\*m(2,2)-m(1,2)\*m(1,2)

d = m(1,1)\*a11+m(1,2)\*a12+m(1,3)\*a13

N(1,1) = a11/d

N(1,2) = a12/d

N(1,3) = a13/d

N(2,2) = a22/d

N(2,3) = a23/d

N(3,3) = a33/d

N(2,1) = N(1,2)

N(3,2) = N(2,3)

N(3,1) = N(1,3)

**return**

**end** **function** inverse2

**function** mbr(x)

**real**, intent(in) :: x

**real** :: mbr

mbr = max(0.0, x)

**return**

**end** **function** mbr

**function** f1(strain, EA, SAT, GAB, GCA, SFS)

**real**, **dimension**(6), intent(in) :: strain

**real**, intent(in) :: EA, SAT, GAB, GCA, SFS

**real** :: f1

f1 = (EA\*mbr(strain(1))/SAT)\*\*2 + &

(GAB\*\*2\*strain(4)\*\*2+GCA\*\*2\*strain(5)\*\*2)/SFS\*\*2

**return**

**end** **function** f1

**function** f2(strain, EA, EB, EC, SAC)

**real**, **dimension**(6), intent(in) :: strain

**real**, intent(in) :: EA, EB, EC, SAC

**real** :: f2, ea\_

ea\_ = -strain(1)-mbr(-EC\*strain(3)-EB\*strain(2))/2./EA

f2 = (EA\*mbr(ea\_)/SAC)\*\*2

**return**

**end** **function** f2

**function** f3(strain, EC, SFC)

**real**, **dimension**(6), intent(in) :: strain

**real**, intent(in) :: EC, SFC

**real** :: f3

f3 = (EC\*mbr(-strain(3))/SFC)\*\*2

**return**

**end** **function** f3

**function** f4(strain, EB, SBC)

**real**, **dimension**(6), intent(in) :: strain

**real**, intent(in) :: EB, SBC

**real** :: f4

f4 = (EB\*mbr(-strain(2))/SBC)\*\*2

**return**

**end** **function** f4

**function** f5(strain, EB, SBT, GBC, SBC0, GAB, SAB0, PHIC)

**real**, **dimension**(6), intent(in) :: strain

**real**, intent(in) :: EB, SBT, GBC, SBC0, GAB, SAB0, PHIC

**real** :: f5, SRB

SRB = EB\*TAN(PHIC\*PI/180.)\*mbr(-strain(2))

f5 = (EB\*mbr(strain(2))/SBT)\*\*2+&

(GBC\*strain(6)/(SBC0+SRB))\*\*2+&

(GAB\*strain(4)/(SAB0+SRB))\*\*2

**return**

**end** **function** f5

**function** f6(strain, EC, SCT, GBC, SBC0, GCA, SCA0, PHIC, S)

**real**, **dimension**(6), intent(in) :: strain

**real**, intent(in) :: EC, SCT, GBC, SBC0, GCA, SCA0, PHIC, S

**real** :: f6, SRC

SRC = EC\*TAN(PHIC\*PI/180.)\*mbr(-strain(3))

f6 = S\*\*2\*((EC\*mbr(strain(3))/SCT)\*\*2+&

(GBC\*strain(6)/(SBC0+SRC))\*\*2+&

(GCA\*strain(5)/(SCA0+SRC))\*\*2)

**return**

**end** **function** f6

**end** **module** mat\_msc\_functions

**function** MatUserAxes (nsh, nip, ncmr, nhv, cmr, strainVelD, strainVelF, strain, stress, stressOld, pu, eu, epsp,&

rou, divU, ts, loc, GM, Km, p\_cut, newTau, tt, time, a, b, c) result(eraseFlag)

!DEC$ ATTRIBUTES DLLEXPORT, ALIAS:"MATUSERAXES" :: MATUSERAXES

**use** mat\_msc\_functions

**implicit** none

!=======входные/выходные параметры=======!

**integer**(4), intent(in) :: nsh

**integer**(4), intent(in) :: nip

**integer**(4), intent(in) :: nhv

**integer**(4), intent(in) :: ncmr

**real**, **dimension** (ncmr), intent(in) :: cmr

**real**, **dimension** (6), intent(in) :: strainVelD

**real**, **dimension** (6), intent(in) :: strainVelF

**real**, **dimension** (6), intent(in) :: strain

**real**, **dimension** (6), intent(inout) :: stress

**real**, **dimension** (6), intent(in) :: stressOld

**real**, **dimension**(3,3):: SS, CC

**real**, **dimension**(3):: CCC

**real**, intent(inout) :: pu

**real**, intent(in) :: eu

**real**, intent(inout) :: epsp

**real**, intent(in) :: rou

**real**, intent(in) :: divU

**real**, intent(in) :: ts

**real**, **dimension**(nhv), intent(inout) :: loc

**real**, intent(inout) :: GM

**real**, intent(inout) :: Km

**real**, intent(inout) :: p\_cut

**logical**, intent(inout) :: newTau

**real**(8), intent(in) :: tt

**real**, intent(in) :: time

**real**, **dimension** (3), intent(in) :: a

**real**, **dimension** (3), intent(in) :: b

**real**, **dimension** (3), intent(in) :: c

**logical** :: eraseFlag

!===========рабочие переменные ===========!

**real** ::divU\_

**logical**, **external** :: hardElasPlasKin

**real** :: cmr1(6)

**real** :: E1, E2, E3, G12, G23, G31

**real** :: gam

**real** :: PR12, PR13, PR23, PR21, PR31, PR32

**real** :: SAT, SAC, SBT, SBC, SCT, SFC, SFS

**real** :: S12, S23, S31, SFFC

**integer** :: AMODEL

**real** :: PHIC, E\_LIMT, S\_DELM, OMGMX, ECRSH

**real** :: EEXPN, CERATE1, CERATE2, CERATE3, CERATE4

**real** :: AM1, AM2, AM3, AM4

**real** :: f1\_, f2\_, f3\_, f4\_, f5\_, f6\_

**real** :: f7\_, f8\_, f9\_, f10\_, f11\_, f12\_, f13\_

**real** :: r1, r2, r3, r4, r5, r6, dr2, dr9, dr10

**real** :: r7, r8, r9, r10, r11, r12, r13

**real** :: w1, w2, w3, w4, w5, w6, w7, w8, w9

**real** :: w10, w11, w12, w13, KE1, KE2

**real**, **dimension** (3,3) :: t\_m

**real**, **dimension** (3, 3):: stressL

**real**, **dimension** (3, 3) :: strainL

**real**, **dimension** (3, 3) :: strainVelFL

**real**, **dimension** (6):: deps

**real**, **dimension** (6):: eps

**real** :: FailType(8)

**real** :: ev ! объемная деформация

!------------------ program code ------------------!

!read(\*,\*)

!

E1 = Cmr(1)

E2 = Cmr(2)

E3 = Cmr(3)

PR21 = Cmr(4)

PR31 = Cmr(5)

PR32 = Cmr(6)

G12 = Cmr(7)

G23 = Cmr(8)

G31 = Cmr(9)

PR12 = E1/E2\*PR21

PR13 = E1/E3\*PR31

PR23 = E2/E3\*PR32

SAT = Cmr(10)

SAC = Cmr(11)

SBT = Cmr(12)

SBC = Cmr(13)

SCT = Cmr(14)

SFC = Cmr(15)

SFS = Cmr(16)

S12 = Cmr(17)

S23 = Cmr(18)

S31 = Cmr(19)

SFFC = Cmr(20)

AMODEL = Cmr(21)

PHIC = Cmr(22)

E\_LIMT = Cmr(23)

S\_DELM = Cmr(24)

OMGMX = min(Cmr(25), 0.999)

ECRSH = Cmr(26)

EEXPN = Cmr(27)

CERATE1 = Cmr(28)

CERATE2 = Cmr(33)

CERATE3 = Cmr(34)

CERATE4 = Cmr(35)

AM1 = Cmr(29)

AM2 = Cmr(30)

AM3 = Cmr(31)

AM4 = Cmr(32)

!------------------------------------

newTau = .false.

**if** (nsh == 0) **then**

loc(1:6) = 0.0d0

loc(7:19) = 1.0d0

loc(20:21) = 0.0d0

gam = 1.0d0/(1.0d0-PR21\*PR12-PR32\*PR23-PR13\*PR31-2.0d0\*PR21\*PR32\*PR13)

Km = max(E1\*(1.0d0-PR32\*PR23)\*gam, E2\*(1.0d0-PR31\*PR13)\*gam, E3\*(1.0d0-PR21\*PR12)\*gam)

newTau = .true.

**end** **if**

eraseFlag = .false.

t\_m(1, 1) = a(1)

t\_m(1, 2) = a(2)

t\_m(1, 3) = a(3)

t\_m(2, 1) = b(1)

t\_m(2, 2) = b(2)

t\_m(2, 3) = b(3)

t\_m(3, 1) = c(1)

t\_m(3, 2) = c(2)

t\_m(3, 3) = c(3)

strainL(1, 1) = strain(1)

strainL(2, 2) = strain(2)

strainL(3, 3) = strain(3)

strainL(1, 2) = strain(4)

strainL(1, 3) = strain(5)

strainL(2, 3) = strain(6)

strainL(2, 1) = strainL(1, 2)

strainL(3, 1) = strainL(1, 3)

strainL(3, 2) = strainL(2, 3)

strainL = matmul(matmul((t\_m),strainL),transpose(t\_m))

eps(1) = strainL(1, 1)

eps(2) = strainL(2, 2)

eps(3) = strainL(3, 3)

eps(4) = strainL(1, 2)\*2

eps(5) = strainL(1, 3)\*2

eps(6) = strainL(2, 3)\*2

strainVelFL(1, 1) = strainVelF(1)

strainVelFL(2, 2) = strainVelF(2)

strainVelFL(3, 3) = strainVelF(3)

strainVelFL(1, 2) = strainVelF(4)

strainVelFL(1, 3) = strainVelF(5)

strainVelFL(2, 3) = strainVelF(6)

strainVelFL(2, 1) = strainVelFL(1, 2)

strainVelFL(3, 1) = strainVelFL(1, 3)

strainVelFL(3, 2) = strainVelFL(2, 3)

strainVelFL = matmul(matmul((t\_m),strainVelFL),transpose(t\_m))

deps(1) = strainVelFL(1, 1)

deps(2) = strainVelFL(2, 2)

deps(3) = strainVelFL(3, 3)

deps(4) = strainVelFL(1, 2)\*2

deps(5) = strainVelFL(1, 3)\*2

deps(6) = strainVelFL(2, 3)\*2

! Расчет поврежденности

w1 = 0.0

w2 = 0.0

w3 = 0.0

w4 = 0.0

w5 = 0.0

w6 = 0.0

dr2 = 0.0

f1\_ = f1(eps, E1, SAT, G12, G31, SFS)

**if** (f1\_.GT.loc(7)\*\*2) **then**

r1 = dsqrt(f1\_)

loc(7) = r1

w1 = 1.0d0 - dexp((1-r1\*\*AM1)/AM1)

**end** **if**

f2\_ = f2(eps, E1, E2, E3, SAC)

**if** (f2\_.GT.loc(8)\*\*2) **then**

r2 = dsqrt(f2\_)

dr2 = r2-loc(8)

loc(8) = r2

w2 = 1.0d0 - dexp((1-r2\*\*AM1)/AM1)

**end** **if**

f3\_ = f3(eps, E3, SFC)

**if** (f3\_.GT.loc(9)\*\*2) **then**

r3 = dsqrt(f3\_)

loc(9) = r3

w3 = 1.0d0 - dexp((1-r3\*\*AM3)/AM3)

**end** **if**

f4\_ = f4(eps, E2, SBC)

**if** (f4\_.GT.loc(10)\*\*2) **then**

r4 = dsqrt(f4\_)

loc(10) = r4

w4 = 1.0d0 - dexp((1-r4\*\*AM2)/AM2)

**end** **if**

f5\_ = f5(eps, E2, SBT, G23, Cmr(18), G12, Cmr(17), PHIC)

**if** (f5\_.GT.loc(11)\*\*2) **then**

r5 = dsqrt(f5\_)

loc(11) = r5

w5 = 1.0d0 - dexp((1-r5\*\*AM4)/AM4)

**end** **if**

f6\_ = f6(eps, E3, SCT, G23, Cmr(18), G31, Cmr(19), PHIC, S\_DELM)

**if** (f6\_.GT.loc(12)\*\*2) **then**

r6 = dsqrt(f6\_)

loc(12) = r6

w6 = 1.0d0 - dexp((1-r6\*\*AM4)/AM4)

**end** **if**

loc(1) = max(loc(1), w1, w2, w3)

loc(1) = min(loc(1), OMGMX)

loc(2) = max(loc(2), w3, w4, w5)

loc(2) = min(loc(2), OMGMX)

loc(3) = max(loc(3), w3, w6)

loc(3) = min(loc(3), OMGMX)

loc(4) = max(loc(4), w1, w2, w3, w4, w5)

loc(4) = min(loc(4), OMGMX)

loc(6) = max(loc(6), w3, w4, w5, w6)

loc(6) = min(loc(6), OMGMX)

loc(5) = max(loc(5), w1, w2, w3, w6)

loc(5) = min(loc(5), OMGMX)

**end** **if**

KE1 = 1.0d0-loc(1)

KE2 = 1.0d0-loc(2)

3 **if** (loc(20).GT.0 .and. eps(1).LT.0) **then**

KE1 = -min(abs(eps(1)\*loc(20)), SAC\*SFFC)/eps(1)/E1

**end** **if**

**if** (loc(21).GT.0 .and. eps(2).LT.0) **then**

KE2 = -min(abs(eps(2)\*loc(21)), SBC\*SFFC)/eps(2)/E2

**end** **if**

! Расчет матрицы податливости

1 SS(1,1) = 1.0d0/E1/KE1 !(1-loc(1))

SS(1,2) = -PR21/E2

SS(1,3) = -PR31/E3

SS(2,1) = SS(1,2)

SS(2,2) = 1.0d0/E2/KE2 !(1-loc(2))

SS(2,3) = -PR32/E3

SS(3,1) = SS(1,3)

SS(3,2) = SS(2,3)

SS(3,3) = 1.0d0/E3/(1.0d0-loc(3))

! Расчет матрицы жесткости

CC = inverse2(SS)

!write(1,"(4E14.3)") time, eps(1), eps(2), eps(3)

CCC(1) = G12\*(1.0d0-loc(4))

CCC(2) = G31\*(1.0d0-loc(5))

CCC(3) = G23\*(1.0d0-loc(6))

! Расчет напряжений

stress(1) = CC(1,1)\*eps(1) + CC(1,2)\*eps(2) + CC(1,3)\*eps(3)

stress(2) = CC(2,1)\*eps(1) + CC(2,2)\*eps(2) + CC(2,3)\*eps(3)

stress(3) = CC(3,1)\*eps(1) + CC(3,2)\*eps(2) + CC(3,3)\*eps(3)

stress(4) = CCC(1)\*eps(4)

stress(5) = CCC(2)\*eps(5)

stress(6) = CCC(3)\*eps(6)

! ОСТАТОЧНАЯ ПРОЧНОСТЬ!!!

**if** (AMODEL.EQ.1) **then**

! Трение

**if** (abs(deps(4)).GT.1e-12 .and. loc(4).GT.(0.5\*OMGMX)) **then**

stress(4) = SIGN(max(abs(stress(4)), mbr(-eps(2))\*E2\*TAN(PHIC\*PI/180.)), stress(4))

**end** **if**

**if** (abs(deps(5)).GT.1e-12 .and. loc(5).GT.(0.5\*OMGMX)) **then**

stress(5) = SIGN(max(abs(stress(5)), mbr(-eps(3))\*E3\*TAN(PHIC\*PI/180.)), stress(5))

**end** **if**

**if** (abs(deps(6)).GT.1e-12 .and. loc(6).GT.(0.5\*OMGMX)) **then**

stress(6) = SIGN(max(abs(stress(6)), &

mbr(-eps(2))\*E2\*TAN(PHIC\*PI/180.), &

mbr(-eps(3))\*E3\*TAN(PHIC\*PI/180.)), &

stress(6))

**end** **if**

! Сжатие

**if** (loc(20).EQ.0.0) **then**

**if** ((dr2.GT.0) .and. (eps(1).LT.0) .and. (abs(E1\*KE1\*eps(1)).LE.(SAC\*SFFC))) **then**

loc(20) = -SAC\*SFFC/eps(1)

**goto** 3

**end** **if**

**end** **if**

**end** **if**

! Обратно в глобальную

stressL(1, 1) = stress(1)

stressL(2, 2) = stress(2)

stressL(3, 3) = stress(3)

stressL(1, 2) = stress(4)

stressL(2, 1) = stress(4)

stressL(1, 3) = stress(5)

stressL(3, 1) = stress(5)

stressL(2, 3) = stress(6)

stressL(3, 2) = stress(6)

stressL = matmul(matmul(transpose(t\_m),stressL),(t\_m))

stress(1) = stressL(1, 1)

stress(2) = stressL(2, 2)

stress(3) = stressL(3, 3)

stress(4) = stressL(1, 2)

stress(5) = stressL(1, 3)

stress(6) = stressL(2, 3)

2 **return**

**end** **function** MatUserAxes

Приложение 2 – Текст программного макета материала (текущая реализцаия)

# coding: utf8

**import** numpy **as** np

**from** copy **import** copy

**from** collections **import** defaultdict

**def** inverse(MM):

# function for inverting 3x3 matrix

N = np.zeros((3, 6), order='F')

N[:3, :3] = MM

**for** i **in** **range**(3):

N[i, i**+**3] = 1.0

**for** i **in** **range**(3):

tmp = N[i, i]

N[i, :] = N[i, :]**/**tmp

**for** j **in** **range**(i**+**1, 3):

tmp = N[j, i]

N[j, :] = N[j, :]**-**N[i, :]**\***tmp

**for** i **in** **range**(2, 0, **-**1):

**for** j **in** **range**(i**-**1, **-**1, **-**1):

tmp = N[j, i]

N[j, :] = N[j, :]**-**N[i, :]**\***tmp

**return** N[:, 3:]

**def** inverse2(M):

**return** np.linalg.inv(M)

# M = np.array([[2., 5, 7], [5, 2, 0], [7, 0, 2]])

# print(M)

# print(np.linalg.inv(M))

# print(inverse(M))

**def** mbr(x): **return** 0 **if** x **<=** 0 **else** x

**def** f1(Cmr, strains):

**return** (Cmr[1]**\***mbr(strains[0])**/**Cmr[10])**\*\***2 **+** \

(Cmr[7]**\*\***2**\***strains[3]**\*\***2**+**Cmr[9]**\*\***2**\***strains[4]**\*\***2)**/**Cmr[16]**\*\***2

**def** f2(Cmr, strains): # , EA, EB, EC, SAC)

ea\_ = **-**strains[0]**-**mbr(**-**Cmr[3]**\***strains[2]**-**Cmr[2]**\***strains[1])**/**2.**/**Cmr[1]

**return** (Cmr[1]**\***mbr(ea\_)**/**Cmr[11])**\*\***2

**def** f3(Cmr, strains):

**return** (Cmr[3]**\***mbr(**-**strains[2])**/**Cmr[15])**\*\***2

**def** f4(Cmr, strains):

**return** (Cmr[2]**\***mbr(**-**strains[1])**/**Cmr[13])**\*\***2

**def** f5(Cmr, strains):

EB = Cmr[2]

SBT = Cmr[12]

GBC = Cmr[8]

SBC0 = Cmr[18]

GAB = Cmr[7]

SAB0 = Cmr[17]

PHIC = Cmr[22]

SRB = EB**\***np.tan(PHIC**\***np.pi**/**180.)**\***mbr(**-**strains[1])

**return** (EB**\***mbr(strains[1])**/**SBT)**\*\***2 **+**\

(GBC**\***strains[5]**/**(SBC0**+**SRB))**\*\***2 **+**\

(GAB**\***strains[3]**/**(SAB0**+**SRB))**\*\***2

**def** f6(Cmr, strains):

EC = Cmr[3]

SCT = Cmr[14]

GBC = Cmr[8]

SBC0 = Cmr[18]

GCA = Cmr[9]

SCA0 = Cmr[19]

PHIC = Cmr[22]

S = Cmr[24]

SRC = EC**\***np.tan(PHIC**\***np.pi**/**180.)**\***mbr(**-**strains[2])

**return** S**\*\***2**\***((EC**\***mbr(strains[2])**/**SCT)**\*\***2 **+**

(GBC**\***strains[5]**/**(SBC0**+**SRC))**\*\***2 **+**

(GCA**\***strains[4]**/**(SCA0**+**SRC))**\*\***2)

loc = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 0.0, 0.0]

strains = np.zeros(6)

sm = None

em = None

**def** mat\_msc(Cmr: **list**, de=np.zeros(6), old\_stress=np.zeros(6), stress\_state=None, step=1):

# в макет передаются приращения деформаций на шаге интегрирования de

# массив параметров материала Cmr

# компоненты напряжений с предыдущего шага old\_stresses

# индекс одномерного напряженного состояния stress\_state.

# если stress\_state не задан или == None, то считается

# что в элементарном объеме задан закон изменения всех компонент тензора деформаций

# loc - параметры состояния материала

**global** loc, strains, sm, em

**if** step **==** 0:

loc = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 0.0, 0.0]

strains = np.zeros(6)

sm = defaultdict(**lambda**: **list**([0]))

em = defaultdict(**lambda**: **list**([0]))

e = strains**+**de

E1 = Cmr[1]

E2 = Cmr[2]

E3 = Cmr[3]

nu21 = Cmr[4]

nu31 = Cmr[5]

nu32 = Cmr[6]

G12 = Cmr[7]

G23 = Cmr[8]

G13 = Cmr[9]

w1 = 0

w2 = 0

w3 = 0

w4 = 0

w5 = 0

w6 = 0

dr2 = 0

# Проверка критериев прочности

f1\_ = f1(Cmr, e)

**if** f1\_ **>** loc[7]**\*\***2:

r = np.sqrt(f1\_)

loc[7] = r

w1 = 1 **-** np.exp((1**-**r**\*\***Cmr[29])**/**Cmr[29])

f2\_ = f2(Cmr, e)

**if** f2\_ **>** loc[8]**\*\***2:

r = np.sqrt(f2\_)

dr2 = r**-**loc[8]

loc[8] = r

w2 = 1 **-** np.exp((1**-**r**\*\***Cmr[29])**/**Cmr[29])

f3\_ = f3(Cmr, e)

**if** f3\_ **>** loc[9]**\*\***2:

r = np.sqrt(f3\_)

loc[9] = r

w3 = 1 **-** np.exp((1**-**r**\*\***Cmr[31])**/**Cmr[31])

f4\_ = f4(Cmr, e)

**if** f4\_ **>** loc[10]**\*\***2:

r = np.sqrt(f4\_)

loc[10] = r

w4 = 1 **-** np.exp((1**-**r**\*\***Cmr[30])**/**Cmr[30])

f5\_ = f5(Cmr, e)

**if** f5\_ **>** loc[11]**\*\***2:

r = np.sqrt(f5\_)

loc[11] = r

w5 = 1 **-** np.exp((1**-**r**\*\***Cmr[32])**/**Cmr[32])

f6\_ = f6(Cmr, e)

**if** f6\_ **>** loc[12]**\*\***2:

r = np.sqrt(f6\_)

loc[12] = r

w6 = 1 **-** np.exp((1**-**r**\*\***Cmr[32])**/**Cmr[32])

# Расчет параметров поврежденности

loc[1] = **min**(**max**([loc[1], w1, w2, w3]), Cmr[25])

loc[2] = **min**(**max**([loc[2], w3, w4, w5]), Cmr[25])

loc[3] = **min**(**max**([loc[3], w3, w6]), Cmr[25])

loc[4] = **min**(**max**([loc[4], w1, w2, w3, w4, w5]), Cmr[25])

loc[6] = **min**(**max**([loc[6], w3, w4, w5, w6]), Cmr[25])

loc[5] = **min**(**max**([loc[5], w1, w2, w3, w6]), Cmr[25])

ke1 = 1**-**loc[1]

ke2 = 1**-**loc[2]

**if** loc[20] **>** 0 **and** e[0] **<** 0:

ke1 = **-min**(**abs**(e[0]**\***loc[20]), Cmr[20]**\***Cmr[11])**/**E1**/**e[0]

**if** loc[21] **>** 0 **and** e[1] **<** 0:

ke2 = **-min**(**abs**(e[1]**\***loc[21]), Cmr[20]**\***Cmr[13])**/**E2**/**e[1]

# Расчет матрицы податливости

S = np.zeros((3, 3))

S[0, 0] = 1.0**/**E1**/**ke1

S[0, 1] = **-**nu21**/**E2

S[0, 2] = **-**nu31**/**E3

S[1, 0] = S[0, 1]

S[1, 1] = 1.0**/**E2**/**ke2

S[1, 2] = **-**nu32**/**E3

S[2, 0] = S[0, 2]

S[2, 1] = S[1, 2]

S[2, 2] = 1.0**/**E3**/**(1**-**loc[3])

# Матрица жесткости

D = inverse(S)

# Расчет напряжений

**if** stress\_state **!=** None:

**if** **type**(stress\_state) **==** **list** **and** **len**(stress\_state) **==** 2:

# только нормальные напряжения, индексы которых содержатся в списке

# stress\_state, отличны от нуля. Оставшееся нормальное напряжение

# зануляется и из этого условия определяется величина соответствующей

# компоненты деформации

idx = (**set**([0, 1, 2])**-set**(stress\_state)).pop()

e[idx] = **-** (D[idx, stress\_state[0]]**\***e[stress\_state[0]] **+**

D[idx, stress\_state[1]]**\***e[stress\_state[1]])**/**D[idx, idx]

**if** **type**(stress\_state) **==** **int**:

# если задан параметр stress\_state в виде числа, обозначающего

# индекс нормального напряжения, то только это нормальное напряжение

# отлично от 0. Считается, что остальные нормальные напряжения

# равны 0 и из этого условия определяются соответствующие компоненты

# нормальных деформаций

idxs = **list**(**set**([0, 1, 2])**-set**([stress\_state]))

b = [**-**D[idxs[0], stress\_state]**\***e[stress\_state],

**-**D[idxs[1], stress\_state]**\***e[stress\_state]]

a = np.zeros((2, 2))

a[0][0] = D[idxs[0], idxs[0]]

a[0][1] = D[idxs[0], idxs[1]]

a[1][0] = D[idxs[1], idxs[0]]

a[1][1] = D[idxs[1], idxs[1]]

ee = np.linalg.solve(a, b)

e[idxs[0]] = ee[0]

e[idxs[1]] = ee[1]

stress = np.zeros(6)

**for** i **in** **range**(3):

**for** j **in** **range**(3):

stress[i] **+**= D[i, j]**\***e[j]

stress[3] = G12**\***e[3]**\***(1**-**loc[4])

stress[4] = G13**\***e[4]**\***(1**-**loc[5])

stress[5] = G23**\***e[5]**\***(1**-**loc[6])

# Остаточная прочность

**if** Cmr[21] **==** 1:

# Трение

**if** (**abs**(de[3]) **>** 1***e*-**12 **and** loc[4] **>** (0.5**\***Cmr[25])):

stress[3] = np.sign(stress[3])**\*max**(**abs**(stress[3]),

mbr(**-**e[1])**\***E2**\***np.tan(Cmr[22]**\***np.pi**/**180.))

**if** (**abs**(de[4]) **>** 1***e*-**12 **and** loc[5] **>** (0.5**\***Cmr[25])):

stress[4] = np.sign(stress[4])**\*max**(**abs**(stress[4]),

mbr(**-**e[2])**\***E3**\***np.tan(Cmr[22]**\***np.pi**/**180.))

**if** (**abs**(de[5]) **>** 1***e*-**12 **and** loc[6] **>** (0.5**\***Cmr[25])):

stress[5] = np.sign(stress[5])**\*max**(**abs**(stress[5]),

mbr(**-**e[1])**\***E2 **\***

np.tan(Cmr[22]**\***np.pi**/**180.),

mbr(**-**e[2])**\***E3 **\***

np.tan(Cmr[22]**\***np.pi**/**180.)

)

# Сжатие

**if** loc[20] **==** 0.0:

**if** ((dr2 **>** 0) **and** (e[0] **<** 0) **and** (**abs**(E1**\***ke1**\***e[0]) **<=** Cmr[11]**\***Cmr[20])):

loc[20] = **-**Cmr[11]**\***Cmr[20]**/**e[0]

stress[0] = **-**Cmr[11]**\***Cmr[20] # E1\*ke1

strains[:6] = e

sm['x'].append(stress[0])

sm['y'].append(stress[1])

sm['z'].append(stress[2])

sm['xy'].append(stress[3])

sm['yz'].append(stress[5])

sm['zx'].append(stress[4])

em['x'].append(strains[0])

em['y'].append(strains[1])

em['z'].append(strains[2])

em['xy'].append(strains[3])

em['yz'].append(strains[5])

em['zx'].append(strains[4])