

الحمد لله الذي
خلقنا من
الحمم



دانشکده مهندسی مکانیک و مکاترونیک
پایان نامه کارشناسی ارشد مهندسی هوافضا - سازه های هوایی

پیش بینی رفتار خزش کامپوزیت های پلیمری تک جهته به روش میکرو مکانیکی

نگارنده: سحر ابراهیمی

استادان راهنما

دکتر مهدی قنادهتوئی

دکتر حسین توننده جانی

مهرماه ۱۳۹۹

در این صفحه صورت جلسه دفاع را قرار دهید. لازم است پس از صحافی این صفحه مجدداً توسط دانشکده مهر گردد و استاد راهنما با امضای خود اصلاحات پایان نامه را تایید کند.

تقدیم اثر

تقدیم به پدرم

کوهی استوار و حامی من در طول تمام زندگی

تقدیم به مادرم

سنگ صوری که الفبای زندگی به من آموخت

تقدیم به همسرم

که در سایه همیاری و بهیلى او به این منظور نائل شدم.

مشکر و قدردانی

مشکر بی پایان مخصوص خدایی است که بشر را آفریده و به او قدرت اندیشیدن عطا کرده و توانایی های بالقوه را در وجود

انسان قرار داده و او را امر به تلاش و کوشش نموده و راهنمایی را برای هدایت بشر فرستاده .

پس از ارادت خاضعانه به درگاه خداوند بی همتا لازم است از اساتید گرامیم جناب آقایان دکتر قناد کهنوی و دکتر تونزنده جانی به خاطر سه صدر و رهنمود های دلسوزانه که مسیر را برایم هموار نمودند مشکر کنم .

تعهدنامه

اینجانب سحر ابراهیمی دانشجوی دوره کارشناسی ارشد رشته مهندسی هوافضا گرایش سازه های هوایی دانشکده مهندسی مکانیک و مکاترونیک دانشگاه صنعتی شاهرود نویسنده پایان نامه پیش بینی رفتار خزش کامپوزیت های پلیمری تک جهته به روش میکرومکانیکی تحت راهنمایی دکتر مهدی قنادهتوئی و دکتر حسین توزنده جانی متعهد می شوم.

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود . استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

استفاده از مواد کامپوزیتی در طراحی سازه‌های اصلی و باربر در صنایع متعدد در حال گسترش است. در مواد کامپوزیتی زمینه پلیمری به دلیل رفتار ویسکوالاستیک فاز زمینه، این فاز دچار پدیده‌ی خزش و آزادسای تنش حتی در دماهای پایین می‌شود. به همین دلیل لحاظ کردن پدیده‌ی فوق در طراحی سازه‌های کامپوزیتی از اهمیت ویژه‌ای برخوردار است.

در این پژوهش ابتدا به معرفی مقیاس‌های مختلف که در تحلیل مواد کامپوزیتی مورد استفاده قرار می‌گیرد، پرداخته شده و سپس مدل‌های مختلف خزشی مورد بحث و بررسی قرار گرفته است. در ادامه مدل میکرومکانیکی جهت بررسی پدیده خزش مورد استفاده قرار گرفته است. مدل‌سازی برای سه مدل تک‌فایبر و مدل تک‌فایبر دارای فازمیانی و مدل توزیع الیاف انجام شده است. به منظور پیش بینی خواص کامپوزیت در درصد حجمی‌های مختلف، مدل‌سازی‌ها با درصد حجمی‌های مختلف انجام شده است. هر سه مدل تحت شرایط مرزی متناوب در مدت زمان ۷۲۰۰ ثانیه و تحت بار ۱ مگاپاسکال و صحت آن توسط نتایج تجربی مورد بررسی قرار گرفت. که در آخر ضرایب کرنش در درصد حجمی‌های مختلف بررسی گردید. در حالت تک‌فایبر با افزایش درصد حجمی الیاف، کرنش کاهش یافته و چون در جهت ۱ خواص ماده بیشتر تابع خواص الیاف است در این جهت تغییرات خواص با دما ناچیز است. در حالت تک‌فایبر دارای فازمیانی پارامتر خرابی d به منظور مدل‌سازی آسیب در فاز میانی مورد استفاده قرار گرفته است. نتایج نشان می‌دهد با افزایش پارامتر خرابی مدول موثر کامپوزیت کاهش می‌یابد و تاثیر این پارامتر در رفتار کاهشی مدول عرضی و برشی بیشتر از مدول طولی کامپوزیت می‌باشد. نتایج به دست آمده در حالت توزیع تصادفی الیاف تطابق خوبی را با نتایج حالت تک فایبر نشان می‌دهد.

واژه‌های کلیدی: کامپوزیت، خزش، میکرومکانیک، درصد حجمی، قانون توان، المان حجمی نماینده

فهرست مطالب

۱	فصل ۱: مقدمه
۲	۱-۱- مقدمه
۲	۲-۱- اجزای یک ماده کامپوزیتی
۴	۱-۲-۱- فاز زمینه
۵	۲-۲-۱- تقویت کننده ها
۷	فصل ۲: مروری بر منابع
۸	۱-۲- مقدمه
۸	۲-۲- انواع مقیاس ها در تحلیل کامپوزیت ها
۸	۱-۲-۲- مقیاس ماکرو
۸	۲-۲-۲- مقیاس مزو
۹	۳-۲-۲- مقیاس میکرو
۱۰	۳-۲- ویسکوالاستیک
۱۱	۴-۲- مدل ویسکوالاستیک برای مواد الاستومری
۱۵	۱-۴-۲- مدل های رئولوژیکال
۲۵	۵-۲- مروری بر ادبیات موضوع
۳۷	۶-۲- نتیجه گیری
۳۹	فصل ۳: روش تحقیق
۴۰	۱-۳- مقدمه
۴۰	۲-۳- بسط مدل قانون توان
۴۲	۳-۳- استفاده از مدل پاور - لا در آباکوس
۴۳	۴-۳- مدل سازی سلول واحد
۴۶	۵-۳- شرایط مرزی متناوب
۴۷	۶-۳- شبکه بندی مدل
۹۴	۷-۳- اسکریپت چیست
۵۰	۸-۳- تولید تصادفی الیاف
۵۵	فصل ۴: نتایج و تفسیر آنها

۵۶	۴-۱- مقدمه
۵۶	۴-۲- صحّه گذاری مدل
۵۷	۴-۳- تعیین خواصّ مدل
۵۸	۴-۴- تاثیر نسبت حجمی بر نتایج
۶۷	۴-۵- تاثیر نسبت حجمی بر نتایج در فاز میانی
۷۴	۴-۶- نتایج توزیع تصادفی الیاف

فصل ۵: جمع‌بندی و پیشنهادات ۸۵

۸۶	۵-۱- محتوا
۸۸	۵-۲- پیشنهادات

مراجع ۸۹

پیوست‌ها ۹۳

فهرست اشکال

- شکل (۱-۱) تصویر کامپوزیت‌های زمینه پلیمری تقویت شده با الیاف [3] ۳
- شکل (۱-۲) انواع طول مقیاس در بررسی خواص سازه کامپوزیتی ۹
- شکل (۲-۲) دو صورت فیزیکی مختلف از رفتار ویسکوالاستیک با توجه به نوع بار اعمالی ۱۱
- شکل (۳-۲) خزش [4] ۱۲
- شکل (۴-۲) مراحل تغییر شکل خزش [4] ۱۳
- شکل (۵-۲) کاربرد کرنش ثابت (b) آزاد سازی تنش [4] ۱۵
- شکل (۶-۲) معرفی مدل‌های مکانیکی [5] ۱۸
- شکل (۱-۳) مثالی از خزشی که فاز اولیه و ثانویه غالب است [7] ۴۱
- شکل (۲-۳) داده‌های آزمایشگاهی مورد استفاده جهت محاسبه ضرائب قانون توانی [42] ۴۲
- شکل (۳-۳) المان حجمی نماینده انتخاب شده ۴۴
- شکل (۴-۳) سلول واحد با درصد حجمی‌های به ترتیب ۰,۱ و ۰,۳ و ۰,۵ و ۰,۷ ۴۵
- شکل (۵-۳) مثالی از نودهای متناوب و غیر متناوب روی صفحه‌های x مثبت و منفی ۴۶
- شکل (۶-۳) شرایط مرزی اعمالی بر سلول واحد ۴۷
- شکل (۷-۳) نمودار حساسیت به مش ۴۸
- شکل (۸-۳) شبکه بندی سلول واحد با سایز ۰,۱ ۴۹
- شکل (۹-۳) نحوه قرار گرفتن تصادفی الیاف در RVE ۵۱
- شکل (۱-۴) مقایسه مدل آزمایشگاهی مرجع [42] با مدل تک‌فایبر ۵۷
- شکل (۲-۴) تاثیر نسبت حجمی بر ϵ_{11} در حالت تک فایبر ۵۹
- شکل (۳-۴) تاثیر نسبت حجمی بر ϵ_{12} در حالت تک فایبر ۵۹
- شکل (۴-۴) تاثیر نسبت حجمی بر ϵ_{22} در حالت تک فایبر ۶۰
- شکل (۵-۴) تاثیر نسبت حجمی بر ϵ_{23} در حالت تک فایبر ۶۰
- شکل (۶-۴) کانتور تنش S11 در حالت تک فایبر ۶۲
- شکل (۷-۴) کانتور تنش S22 در حالت تک فایبر ۶۳
- شکل (۸-۴) کانتور تنش S12 در حالت تک فایبر ۶۵
- شکل (۹-۴) کانتور تنش S23 در حالت تک فایبر ۶۶
- شکل (۴-۱۰) نمودار نرمی خزشی ϵ_{11} و ϵ_{22} و ϵ_{12} و ϵ_{23} در حالت فاز میانی ۶۹
- شکل (۱۱-۴) مقایسه حالت تک‌فایبر با حالت دارای فاز میانی ۷۰

- شکل (۴-۱۲) کانتور تنش S11 در حالت فاز میانی ۷۱
- شکل (۴-۱۳) کانتور تنش S12 در حالت فاز میانی ۷۲
- شکل (۴-۱۴) کانتور تنش S22 در حالت فاز میانی ۷۳
- شکل (۴-۱۵) کانتور تنش S23 در حالت فاز میانی ۷۴
- شکل (۴-۱۶) نمودار نرمی خزشی ϵ_{11} در حالت تولید تصادفی ۷۵
- شکل (۴-۱۷) نمودار نرمی خزشی ϵ_{22} در حالت تولید تصادفی ۷۶
- شکل (۴-۱۸) نمودار نرمی خزشی ϵ_{12} در حالت تولید تصادفی ۷۷
- شکل (۴-۱۹) نمودار نرمی خزشی ϵ_{23} در حالت تولید تصادفی ۷۷
- شکل (۴-۲۰) مقایسه کردن مدل ۴ فایبر با مدل تک فایبر ۷۸
- شکل (۴-۲۱) کانتور تنش S11 در حالت تولید تصادفی ۷۹
- شکل (۴-۲۲) کانتور تنش S22 در حالت تولید تصادفی ۸۰
- شکل (۴-۲۳) کانتور تنش S12 در حالت تولید تصادفی ۸۱
- شکل (۴-۲۴) کانتور تنش S23 در حالت تولید تصادفی ۸۲

فہرست جداول

- جدول (۳-۱) ثابت‌های به دست آمده از برازش منحنی معادله پاور-لا ۴۲
- جدول (۳-۲) خواص الاستیک رزین ۴۳
- جدول (۳-۳) خواص الاستیک الیاف کربن ۴۳

فهرست علائم

S_{ijkl}	درایه‌های ماتریس نرمی
S_0	نرمی خزشی آنی
$\Delta_{S(t)}$	مولفه گذرای نرمی
$\dot{\epsilon}$	نرخ خزش
σ_0	تنش وارده خارجه
T	دمای تست
$\dot{\epsilon}_m$	نرخ کرنش ماتریس
V_f	نسبت حجمی الیاف
λ_f	پارامترهای هندسی الیاف
θ	زاویه قرارگیری الیاف
E_f	مدول الیاف
σ_{uf}	مقاومت نهایی الیاف
E_m	مدول ماتریس
E_i	مدول سطح مشترک الیاف و ماتریس
t_i	ضخامت سطح مشترک
γ	کرنش
$J(t)$	پاسخ کرنش
J_g	ضریب شیشه
J_e	ضریب تعادل
b_1	ضریب ویسکوزیته

M	ضرب سفتی
τ_{ε}	زمان تاخیر
a_1 و b_1	جنس ماده
$\dot{\varepsilon}_{ij}^e$	درایه‌های کرنش الاستیک
$\dot{e}_{ij}$	نرخ تغییر شکل کلی
$\emptyset$	تابع پتانسیل
σ_{ij}	تنش
D_{ij}	تنسور جهت‌گیری
ξ, η, ζ	پارامترهای غیر ایزوتروپیک
T_s	برش عرضی
L_s	برش طولی
T_N	نرمال عرضی
L_N	نرمال طولی
A, m, n	ثابت‌های مواد
E_t	مدول وابسته به زمان
E, E_v	مدول ویسکوالاستیک
E_0	مدول الاستیسیته اولیه مستقل از زمان
$D_0, \Delta D(\varphi)$	ضرایب ویسکوالاستیک خطی
φ	زمان کاهش یافته
g_0, g_1, g_2	پارامترهای غیرخطی وابسته به تنش
a_{σ}	ضرب جابجایی زمان
L_x	فاصله لبه از x

L_y	فاصله لبه از y
N	تعداد الیاف
d	پارامتر آسیب
X, Y	مختصات الیاف
tol	تولرانس اولیه
t	زمان
r	فاصله بین دو نود

فصل ۱:

مقدمه

۱-۱- مقدمه

کامپوزیت‌ها، به طور خاص کامپوزیت‌های پلیمری با الیاف کربن^۱ رشد قابل توجهی در مصارف و کاربردهای صنعت هوافضا داشته‌اند. علت اصلی افزایش استفاده از این مواد استحکام و ضریب صلبیت بالای این مواد در کنار چگالی پایین و قیمت مناسب و وزن بسیار کم آن‌هاست. از کاربردهای این مواد می‌توان به صنایع هوافضا، کشتی‌سازی، مخزن تحت فشار، اتومبیل‌ها و ... اشاره کرد. استفاده از کامپوزیت‌ها از سال‌ها پیش مورد استفاده قرار می‌گرفته است. [1]

۱-۲- اجزای یک ماده‌ی کامپوزیتی

اساسی‌ترین شکل یک ماده‌ی کامپوزیتی، شکلی است که در آن دو جز ترکیب شده‌اند و ماده‌ای را با خواصی که متفاوت از خصوصیات اجزای آن است تولید کرده‌اند. اغلب کامپوزیت‌ها شامل یک ماده‌ی حجمی^۲ است به نام زمینه و یک تقویت کننده با انواع مختلف که برای افزایش سفتی و استحکام زمینه اضافه شده‌است. این تقویت کننده اغلب به شکل الیاف است. [2] کامپوزیت‌ها با توجه به نوع ماتریسشان به سه نوع پلیمری، سرامیکی و فلزی تقسیم می‌شوند.

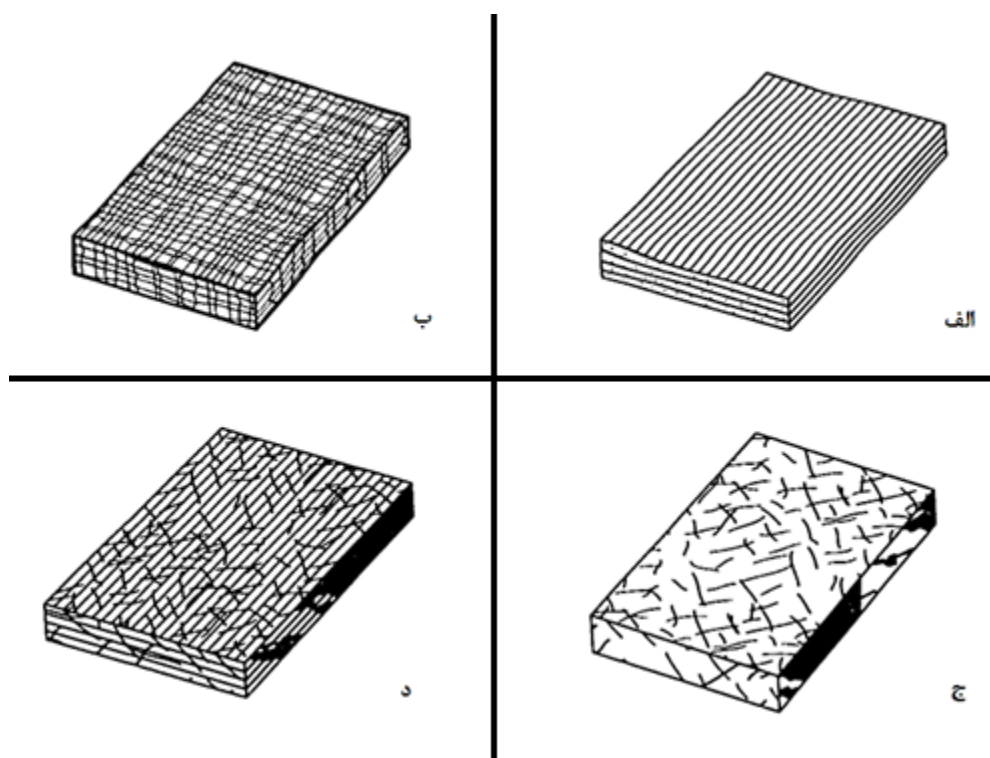
به‌طور کلی کامپوزیت‌های زمینه پلیمری تقویت شده با الیاف را از نقطه نظر نحوه چیدمان الیاف درون زمینه می‌توان به چهار دسته تقسیم‌بندی نمود. این دسته‌بندی شامل کامپوزیت‌های زمینه پلیمری تقویت شده با الیاف پیوسته، الیاف بافته شده، الیاف رشته‌ای و الیاف هیبریدی هستند. دسته اول کامپوزیت‌های تقویت شده با الیاف پیوسته می‌باشند که تصویر نحوه جهت‌گیری الیاف داخل زمینه در شکل (۱-۱ الف) آورده شده است. این نوع جهت‌گیری به صورت گسترده در ساخت قطعات کامپوزیتی مورد استفاده قرار می‌گیرند، با

¹ Carbon-Fiber-Reinforced Polymer (CFRP)

² Bulk

این حال مستعد به لایه لایه شدن و جدایش لایه‌ای می‌باشند.

کامپوزیت‌های تقویت شده با الیاف بافته شد در شکل (۱-۱ ب) آورده شده‌اند. این دسته از مواد کامپوزیتی مستعد به لایه لایه شدن نبوده و نسبت به الیاف پیوسته دارای استحکام کمتری هستند. کامپوزیت‌های تقویت شده با الیاف ناپیوسته به علت ارزانی روش تولید، به طور گسترده در تولیدات انبوه مورد استفاده قرار می‌گیرند و نسبت به نوع تقویت شده با الیاف پیوسته خواص مکانیکی ضعیف‌تری دارند. شکل (۱-۱ ج) کامپوزیت‌های تقویت شده با الیاف هیبریدی نیز شکل (۱-۱ د)، شامل دو یا چند نوع الیاف پیوسته، بافته شده و رشته‌ای از جنس‌هایی مانند شیشه، گرافیت و غیره می‌باشند.



شکل (۱-۱) تصویر کامپوزیت‌های زمینه پلیمری تقویت شده با الف)الیاف پیوسته -ب)الیاف بافته شده -ج)الیاف رشته‌ای - د)الیاف هیبریدی [3]

در میان کامپوزیت‌ها، کامپوزیت‌های پایه پلیمری به علت خواص منحصر به فرد از جمله سبکی، مقاومت به خوردگی و فرایند تولید نسبتاً آسان در صنایع بسیاری مورد توجه است. به علت وجود پلیمر به عنوان ماده‌ی زمینه، این نوع کامپوزیت خواص خزشی متفاوتی نسبت به فلزات و حتی دیگر کامپوزیت‌ها از خود نشان

می‌دهد به طوری که برخی از پلیمرها در دمای محیط هم دچار خزش می‌شوند. یکی از چالش برانگیزترین مسائل در بحث خزش کامپوزیت‌ها، ارائه راهکاری برای پیش‌بینی رفتار خزشی آن‌ها بر اساس مدل‌سازی تئوری به منظور پرهیز از آزمون‌های درازمدت و وقت‌گیر می‌باشد.

راهکارهای متفاوتی برای پیش‌بینی رفتار خزشی کامپوزیت‌های پایه پلیمری ارائه شده است. بسته به نوع کامپوزیتی که مورد مطالعه قرار دارد، این راهکار متفاوت است. روش‌های ارائه شده برای پیش‌بینی خزش کامپوزیت‌های پایه پلیمری را می‌توان به پنج دسته کلی مدل‌سازی تحلیلی، مدل‌سازی تجربی و نیمه تجربی، مدل‌سازی رئولوژیکال، مدل‌سازی اجزای محدود و روش‌های پیش‌بینی با استفاده از قوانین برهم‌نهی (پیش‌بینی بلند مدت) تقسیم کرد که هر کدام مشخصات خاصی دارند. نتایج مطالعات انجام شده نشان می‌دهد که هیچ یک از روش‌های ذکر شده توانایی پیش‌بینی بلندمدت خزش را در تمام شرایط و به طور جامع در مورد کامپوزیت‌های پایه پلیمری ندارد اما بسته به شرایط مسئله هر کدام می‌تواند به دیگر روش‌ها ارجحیت داشته باشد. دلیل این امر را می‌توان در پشتوانه‌ی تئوری هر کدام از این روش‌ها جستجو کرد. در ادامه به توضیح فاز زمینه و تقویت‌کننده‌ها پرداخته می‌شود.

۱-۲-۱- فاز زمینه

خیلی از مواد هنگامی که به صورت الیاف هستند استحکام خیلی خوبی را نشان می‌دهند، اما برای رسیدن به این خواص، الیاف باید به یک زمینه‌ی خوب پیوند زده شوند. زمینه الیاف را از هم جدا می‌کند تا از ساییدگی و تشکیل عیوب سطحی جدید جلوگیری نماید و مانند پلی، الیاف را در محلی نگه دارد. یک زمینه‌ی خوب باید توانایی تغییر شکل تحت بار به کاررفته را دارا باشد و نیرو را به الیاف انتقال دهد و تمرکز تنش را توزیع نماید، الیاف را از صدمات محافظت نموده و از اشاعه ترک در کامپوزیت جلوگیری کند. مطالعه‌ی طبیعی نیروهای پیوندی نشان می‌دهد که به محض اولین بارگذاری، تمایلی به چسبندگی بین تقویت‌کننده و زمینه

که می‌خواهد شکسته شود، وجود دارد. این نیروهای چسبندگی برای جلوگیری از جدایش دو فاز^۱ ضروری است. این نیروهای پیوندی باید به حدّ کافی بزرگ باشند تا از این رویداد جلوگیری کنند همچنین نیروهای پیوندی نقش مهمی در انتقال بار دارد. زمینه معمولاً حدود درجه حرارت کار را مشخص می‌کند. عمومی - ترین کامپوزیت‌های ساخته شده را بر اساس نوع زمینه می‌توان به سه گروه اصلی تقسیم نمود.[2]

پلیمرها به انواع مختلفی تقسیم می‌شوند که این تقسیم بندی می‌تواند بر حسب نوع سنتز، نوع ترکیب شیمیایی و نوع ساختار مولکولی باشد. مهم‌ترین طبقه‌بندی پلیمرها به صورت زیر است.

✓ ترموپلاستیک‌ها

✓ ترموست‌ها

✓ الاستومرها

۱-۲-۲- تقویت کننده‌ها

تقویت کننده‌های کامپوزیت‌های پایه پلیمری به سه دسته تقسیم می‌شوند: [2]

✓ پلیمرهای تقویت شده با الیاف شیشه^۲

✓ پلیمرهای تقویت شده با الیاف کربن^۳

✓ پلیمرهای تقویت شده با الیاف آرامید^۴

بسیاری از کامپوزیت‌ها از ترکیباتی مانند رزین ساخته شده‌اند که رفتاری ویسکوالاستیک را دارا هستند.

¹ Pull-Out

² Glass fiber reinforced polymers (GFRP)

³ Carbon fiber reinforced polymers (CFRP)

⁴ Aramid fibers

یکی از بحث‌های مهم در تحلیل رفتار مکانیکی کامپوزیت‌ها بررسی رفتار خزشی آن‌ها است. روش میکرومکانیکی ابزار بسیار موثری برای مشخص کردن رفتار مواد در کامپوزیت با توجه به رفتار مشخص آن از طریق یک المان حجمی نمونه است.

۱-۳- ساختار پایان‌نامه

در فصل اول به بررسی اجزا یک ماده‌ی کامپوزیتی که شامل فاز زمینه و تقویت کننده‌ها است پرداخته شد. در فصل دوم این پژوهش، به انواع مقیاس‌ها در کامپوزیت‌ها و مدل ویسکوالاستیک و انواع مدل‌های خزش و مقالات مرور شده در این حوزه پرداخته شده است. در فصل سوم با استفاده از یک کد پایتون به بسط مدل قانون توان و اعمال آن و نحوه مدل‌سازی در نرم افزار اجزا محدود آباکوس پرداخته می‌شود. و در فصل چهارم به صحنه گذاری مدل می‌پردازیم و در ادامه نتایج برای سه حالت تک فایبر و تک فایبر دارای فاز میانی و حالت توزیع تصادفی الیاف آورده شده است.

فصل ۲:

مروری بر منابع

۲-۱- مقدمه

در این فصل ابتدا انواع مقیاس‌ها در تحلیل کامپوزیت‌ها را بررسی می‌کنیم. سپس به منابع گذشته و تحقیقات انجام شده بر روی پیش‌بینی رفتار خزش کامپوزیت‌های پلیمری تک‌جهته به روش میکرومکانیکی می‌پردازیم.

۲-۲- انواع مقیاس‌ها در تحلیل کامپوزیت‌ها

رفتار کامپوزیت‌ها در طبیعت به صورت سلسه مراتبی است و در نتیجه شامل طیف مختلفی از طول مقیاس‌ها می‌باشد که باعث تعیین رفتار کلی کامپوزیت‌ها می‌شود. مقیاس‌های مختلف برای کامپوزیت‌های مختلف در بررسی کامپوزیت‌ها با الیاف کربن می‌تواند به صورت زیر طبقه بندی شود.

۲-۲-۱- مقیاس ماکرو

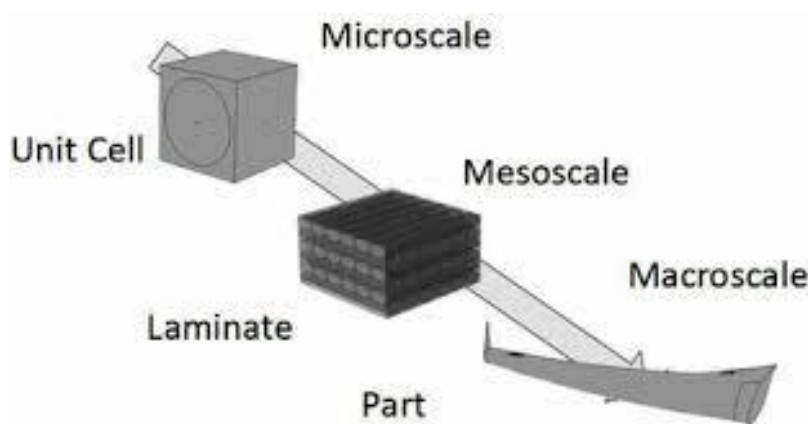
طول مقیاسی متناسب با کلّ سازه که شامل بارگذاری در شرایط مرزی از تحلیل یا تست کلی سازه است. در تحلیل مقیاس ماکرو، خواصّ موثر ماده‌ی همگن شده توسط نرم افزارهای ساختاری برای یافتن تغییرشکل‌هایی که در تعادل با بارگذاری خارجی هستند، استفاده می‌شوند.

۲-۲-۲- مقیاس مزو

مقیاسی میانی است که مربوط به نحوه‌ی ساختمان خاصّ تقویتی از جمله چیدمان لایه‌ها برای کامپوزیت چند لایه و یا نوع کامپوزیت از نظر نحوه بافت می‌باشد.

۲-۳-۲- مقیاس میکرو

برای تعیین خواص مواد مرکب، از خواص اجزای تشکیل دهنده، نسبت حجمی و اثر متقابل بین فازها که مربوط به هندسه میکرو ساختار ماده مرکب است استفاده می‌شود. بنابراین برای تحلیل رفتار حرارتی و مکانیکی ماده مرکب، اثر متقابل مواد تشکیل دهنده و خواص هر کدام از آن‌ها به صورت میکروسکوپی بررسی می‌شود. با مدل‌سازی میکرومکانیک و با اطلاع از خواص ریزساختاری ماده مرکب از جمله خواص الیاف، زمینه و هندسه مدل می‌توان خواص متوسط ماده مرکب را تخمین زد. بنابراین با کمک گرفتن از بررسی میکروسکوپی و میکرومکانیکی یک ماده‌ی چند جنسی می‌توان یک ماده غیر ایزوتروپ همگن معادل برای بررسی‌های میکرومکانیکی ارائه کرد. ماده همگن یا ماده ایزوتروپیک (همسانگرد) ماده‌ای است که اندازه‌گیری خواص در هر جهت یکسان است. المان حجمی نمونه یا تکنیک سلول واحد تکراری بسته به هندسه و شرایط بارگذاری کامپوزیت ساخته می‌شود تا رفتار کلی کامپوزیت و خواص آن از طریق تحلیل المان حجمی نمونه یا سلول واحد با روش‌های تحلیلی یا عددی تعیین شوند.



شکل (۱-۲) انواع طول مقیاس در بررسی خواص سازه کامپوزیتی [4]

۲-۳- ویسکوالاستیک

ویسکوالاستیسیته خاصیتی از مواد است که نشان دهنده‌ی خواص الاستیک و ویسکوز تحت تغییر شکل است. یک ماده ویسکوز رفتار وابسته به زمان از خود نشان می‌دهد به طوری که با اعمال تنش ثابت با نرخ ثابتی دچار تغییر شکل می‌شود و زمانی که بار برداشته می‌شود، ماده حالت اولیه خود را فراموش می‌کند و در حالت تغییر شکل باقی می‌ماند. از طرف دیگر مواد الاستیک با اعمال بار دچار تغییر شکل می‌شوند و حالت اولیه خود را به خاطر می‌آورند یعنی بعد از باربرداری به حالت اولیه خود باز می‌گردند. مواد ویسکوالاستیک دارای المان‌هایی از این دو نوع خاصیت دارند، به طور مثال کرنش وابسته به زمان دارند و دارای حافظه‌ای هستند که به تدریج حذف می‌شود^۱. چنین رفتاری می‌تواند خطی یا غیر خطی باشد. در حقیقت رفتار الاستیک معمولاً نتیجه‌ی کش آمدن اتصالات در راستای صفحات کریستالی در یک جامد منظم هستند و رفتار ویسکوالاستیک نتیجه نفوذ^۲ اتم‌ها یا مولکول‌های یک ماده‌ی آمورف (بی شکل) درون یکدیگر است. این واقعیت درباره‌ی پلیمرها اثبات شده‌است که رفتار آن‌ها تحت بار یا تغییر شکل حتی در دمای اتاق هم وابسته به زمان است. علاوه بر این، پاسخ آن‌ها به بارگذاری یا تغییر شکل در برخی موارد به بار قبلی و تاریخچه دما یا تغییر شکل بستگی دارد. این وابستگی به زمان خودش را به چند فرم نشان می‌دهد، که سه مورد از آن‌ها خزش^۳، بازیابی^۴ و رهاسازی تنش^۵ هستند که خزش به افزایش پیش‌رونده تغییر شکل تحت بار ثابت گفته می‌شود و رهاسازی تنش به کاهش تدریجی تنش تحت تغییر شکل ثابت گفته می‌شود.

دو صورت فیزیکی مهم از رفتار ویسکوالاستیک مواد به چشم می‌خورد. همان‌طور که در شکل (۲-۲) ملاحظه می‌کنید، تیری ویسکوالاستیک تحت دو نوع بار مختلف قرار گرفته‌است.

¹ fading memory

² diffusion

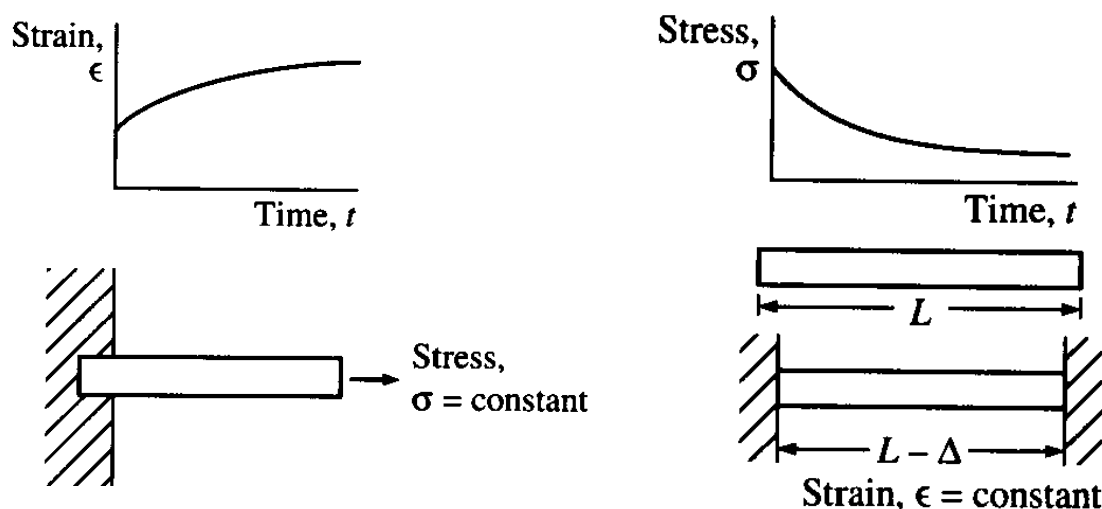
³ creep

⁴ recovery

⁵ stress relaxation

شکل (الف) اگر این تیر تحت کرنش و یا تغییر طول ثابتی قرار بگیرد، تنش حاصل شده رفتاری تابع زمان (آسودگی) از خود نشان می دهد. یعنی اینکه تنش بعد از اعمال تنش الاستیک اولیه کاهش پیدا می کند.

شکل (ب) اگر این تیر تحت تنش ثابت قرار بگیرد، کرنش حاصل شده رفتاری تابع زمان (خزش) از خود نشان می دهد. که رفتار تابع زمان کرنش بعد از کرنش اولیه الاستیک می باشد.



(ب) خزش تحت تنش ثابت

الف (آسودگی تحت کرنش ثابت

شکل (۲-۲) دو صورت فیزیکی مختلف از رفتار ویسکوالاستیک با توجه به نوع بار اعمالی

۲-۴- مدل ویسکوالاستیک برای مواد الاستومری^۱

ماهیت الاستومرها، ویسکوالاستیک است. اکثر مدل های اساسی سازنده الاستومرها بر اساس تئوری های ویسکوالاستیسیته خطی و غیر خطی می باشد. در این بخش مدل های رایج برای توصیف ویسکوالاستیک خطی مورد بحث قرار می گیرد. رفتار مواد ویسکوالاستیک از طریق دو تست مشخص می شود.

➤ تست رهایش تنش

➤ تست کرنش خزشی

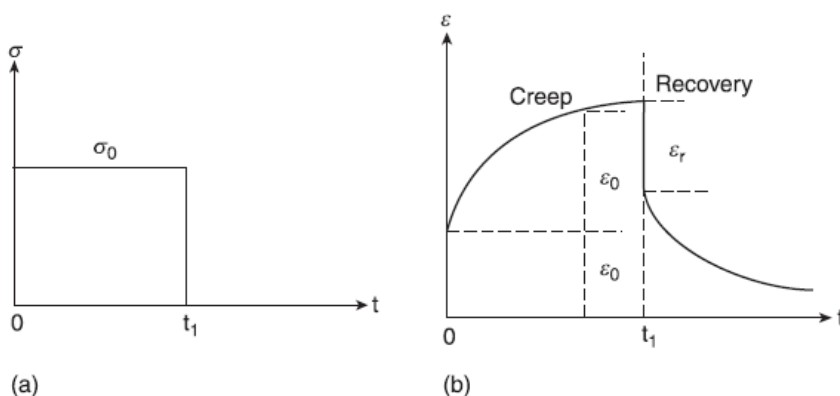
¹ Elastomeric

با توجه به این تست‌ها مدول را از طریق برازش داده‌های آزمایشگاهی که می‌توان برای توصیف چندین پاسخ ویسکوالاستیک این مواد استفاده کرد، استخراج کرد. تست‌های دینامیکی همچنین برای تعریف مدول مربوط به فرکانس انجام می‌شود که یکی از ویژگی‌های مهم مواد ویسکوالاستیک است. مطالعه‌ی رفتار وابسته به زمان مواد در سه بخش خزش، بازیابی^۱ و رهاسازی تنش بررسی می‌شود.

□ خزش

خزش بیان‌گر تغییر شکل آهسته و پیوسته مواد تحت تنش ثابت است. بر خلاف فلزات، پلیمرها حتی در دمای اتاق هم دچار خزش می‌شوند. پاسخ خزش به یک تنش ثابت وارده در زمان $t=0$ در شکل (۳-۲) نشان داده شده است. یک کرنش ϵ_0 متناظر با تنش اعمالی بعد از اعمال تنش مشاهده می‌شود و با یک افزایش پیش‌رونده کرنش همانطور که در شکل (۳-۲) نشان داده شده است، دنبال می‌شود. کرنش کل در هر لحظه از زمان برابر با مجموع کرنش الاستیک و کرنش خزشی است.

$$\epsilon(t) = \epsilon_e + \epsilon_c \quad (۱-۲)$$



شکل (۳-۲) خزش: (a) اعمال تنش ثابت (b) پاسخ کرنش [5]

¹ recovery

به نسبت کرنش کل به تنش ثابت اعمالی نرمی خزشی^۱ می‌گویند که از رابطه‌ی (۲-۲) به‌دست می‌آید.

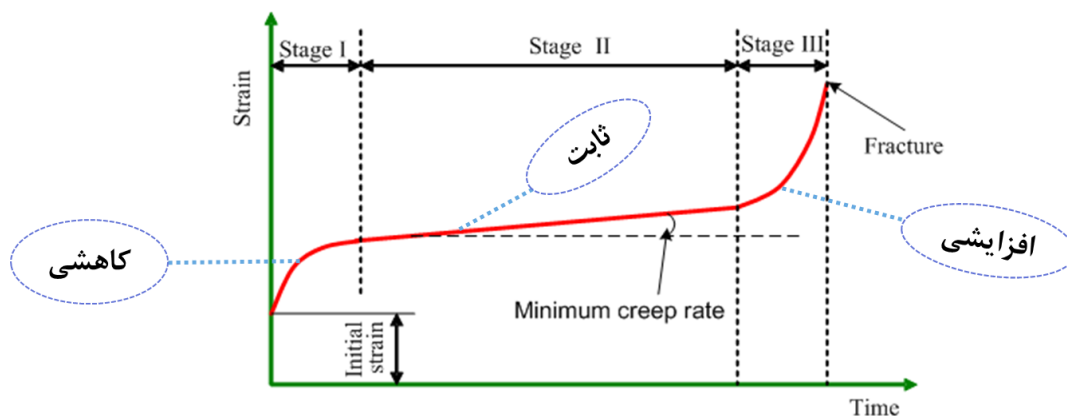
$$S(t) = \frac{\varepsilon(t)}{\sigma_0} \quad (۲-۲)$$

نرمی خزشی در هر نقطه از زمان به صورت رابطه‌ی (۳-۲) تعریف می‌شود.

$$S(t) = S_0 + \Delta S(t) = S(0) + \Delta S(t) \quad (۳-۲)$$

که در آن S_0 نرمی خزشی آنی و $\Delta S(t)$ مولفه گذرای نرمی است.

به طور کلی خزش را می‌توان در سه مرحله توصیف کرد، در مرحله اول ماده دچار تغییر شکل با نرخ کاهشی می‌شود، در مرحله دوم نرخ تغییر شکل تقریباً ثابت است. در مرحله سوم تغییر شکل دارای نرخ افزایشی است تا به نقطه شکست می‌رسد (شکل (۴-۲)).



شکل (۴-۲) مراحل تغییر شکل خزش [5]

فهم رفتار خزش یک ماده در طراحی و ساخت دارای اهمیت زیادی است به طوری که می‌تواند منجر به ناپایداری ابعادی محصول نهایی شود، همچنین شکست در تنش‌هایی که خیلی کمتر از مقاومت کششی

¹ creep compliance

است می تواند رخ دهد.

خزش یک پدیده کاملاً پیچیده است به طوری که به پارامترهای زیادی بستگی دارد به طور مثال رفتار خزش کامپوزیت‌های تقویت‌شده با الیاف بر فاکتورهایی مانند رفتار خزش ماتریس، رفتار شکست و الاستیک الیاف، هندسه و چینش الیاف و خواص بین وجهی الیاف و ماتریس بستگی دارد. به طور کلی نرخ خزش کامپوزیت‌ها را می‌توان با معادله (۴-۲) بیان کرد.

$$\dot{\varepsilon} = f(\sigma_0, T, \dot{\varepsilon}_m, V_f, \lambda_f, \Theta, E_f, \sigma_{uf}, E_m, E_i, t_i) \quad (4-2)$$

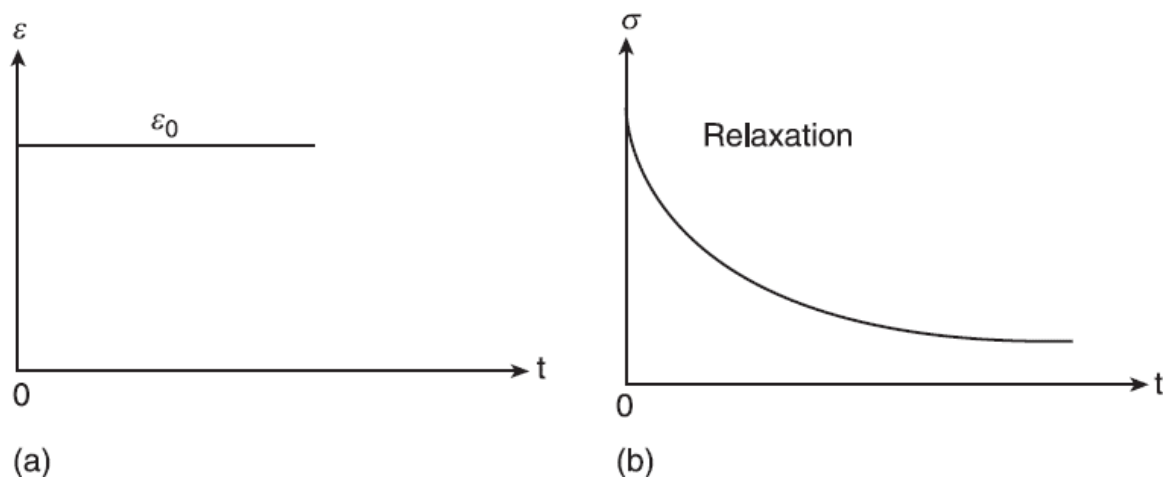
به طوری که پارامترهای درون پرانتز به ترتیب تنش وارده خارجی، دمای تست، نرخ کرنش ماتریس، نسبت حجمی الیاف، پارامترهای هندسی الیاف، زاویه قرار گیری الیاف، مدول الیاف، مقاومت نهایی الیاف، مدول ماتریس، مدول سطح مشترک الیاف و ماتریس، ضخامت سطح مشترک هستند. [5]

□ بازیابی

اگر بار برداشته شود، یک کرنش الاستیک معکوس و به دنبال بازیابی بخشی از کرنش خزشی در یک نرخ کاهشی اتفاق می‌افتد. مقدار کرنش قابل بازیابی وابسته به زمان در حین بازیابی به طور کلی قسمت بسیار کمی از کرنش خزشی وابسته به زمان برای فلزات است، در حالی که برای پلاستیک‌ها ممکن است قسمت بزرگی از کرنش خزشی وابسته به زمان باشد شکل (۳-۲) اگر زمان کافی برای بازیابی در نظر گرفته شود، ممکن است برخی از پلاستیک‌ها بازیابی کامل داشته باشند. بازیابی کرنش خزشی تاخیری نیز نامیده می‌شود. [5]

□ رهاسازی تنش

مواد ویسکوالاستیک که تحت فشار ثابت قرار می‌گیرند، شل می‌شوند شکل (۵-۲a). به طوری که تنش به تدریج کاهش می‌یابد همانطور که در شکل (۵-۲) نشان داده شده است. [5]



شکل (۵-۲) (a) کاربرد کرنش ثابت (b) آزاد سازی تنش [5]

۲-۴-۱- مدل های رئولوژیکال^۱

مطالعات زیادی جهت ارائه مدلی برای رفتار خزش صورت پذیرفته است. مدل های مکانیکی نقش مهمی در مقالات علمی که پیشرفت تاریخی را بوجود آورده اند ایفا می کند. در واقع تئوری های اولیه با کمک این مدل ها بوجود آمده اند که همچنان در مجسم کردن خواص و قوانین تئوری کلی، با استفاده از قانون ترکیب موثر هستند. در ادامه مدل های ساده ای که رابطه ی تنش-کرنش با تابع مواد را پوشش می دهد را آورده ایم.

۲ [۶] مدل نیوتون^۲

دمپر یک المان ویسکوز است که در آن نیرو بر نرخ کشیدگی تقسیم می گردد. این مدل جسمی کاملاً ویسکوز را مدل می کند که از قانون نیوتون پیروی می کند. بنابراین به نام مدل نیوتون نام گذاری شده است. که معادله آن به صورت معادله (۵-۲) می باشد.

¹ Rheological models

² Newton

$$\sigma(t) = b_1 \frac{d\varepsilon}{dt} \quad \text{and} \quad J(t) = \frac{t}{b_1} \quad (5-2)$$

در این حالت ضریب خزش به صورت خطی تعریف می‌گردد و پارامترها به صورت زیر تعریف می‌گردد.

$J(t)$ = پاسخ کرنش به تنش واحد بر اساس تست خزش که به ضریب خزش^۱ اشاره دارد.

تابع $J(t)$ ، در بازه‌ی $0 < t < +\infty$ تابعی غیر نزولی است.

b_1 ضریب ویسکوزیته مربوطه و t زمان می‌باشد.

$\frac{d\varepsilon}{dt}$ نرخ کرنش می‌باشد.

□ مدل کلوین-وویت^۲ [6]

در این مدل فنر و دمپر به صورت موازی به هم متصل بوده و بار اعمالی بین هر دو تقسیم می‌گردد. از این مدل برای تنش متغیر نمی‌توان استفاده کرد و این مدل قادر به مدل‌سازی مناسب و دقیق ویسکوالاستیک مواد نمی‌باشد. روابط تنش-کرنش و ضریب خزش به صورت معادله‌ی (۶-۲) تعریف می‌گردد.

$$\sigma(t) = M\varepsilon(t) + b_1 \frac{d\varepsilon}{dt} \quad (6-2)$$

$$J(t) = J_1 \left[1 - e^{-t/\tau_\varepsilon} \right]$$

$$J_1 = \frac{1}{M} \quad , \quad \tau_\varepsilon = \frac{1}{M}$$

τ_ε به عنوان زمان تاخیر^۳ شناخته می‌شود. و M به عنوان ضریب سفتی است. J_1 ضریب نرمی می‌باشد.

مدل کلوین، مدلی است که یک خزش کرنشی نمایی (وارونپذیر) را نمایش می‌دهد و به عنوان یک المان تاخیری اشاره دارد.

¹ Creep Compliance

² Kelvin-Voigt

³ Retardation time

□ مدل ماکسول [6]^۱

ترکیب یک فنر و دمپر به صورت سری مدل ماکسول را تعریف می‌کند. در ادامه روابط تنش- کرنش و ضریب خزش در این مدل به صورت معادله‌ی (۷-۲) مشخص می‌شود.

$$\sigma(t) + a_1 \frac{d\sigma}{dt} = b_1 \frac{d\varepsilon}{dt} \quad (7-2)$$

$$J(t) = J_g + J_+ t \quad , \quad J_g = \frac{a_1}{b_1}, J_+ = \frac{1}{b_1}$$

a_1 و b_1 مربوط به جنس ماده بوده و از تست خزشی به دست می‌آید.

$J_g = J(0^+)$ ضریب شیشه^۲ و $J_e = J(+\infty)$ ضریب تعادل^۳ نام‌گذاری می‌شوند.

$\sigma(t)$ تنش

$\frac{d\sigma}{dt}$ نرخ تنش

$\frac{d\varepsilon}{dt}$ نرخ کرنش می‌باشند.

مدل ماکسول، مدلی خطی (غیر وارونپذیر) از خزش کرنشی بوده و تحت عنوان المان سستی^۴ شناخته

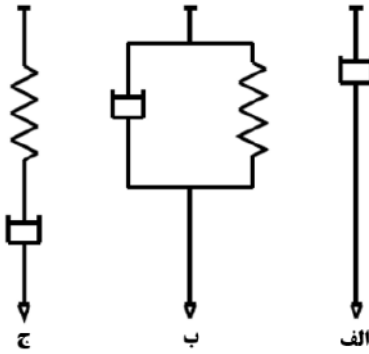
می‌شود. همانطور که در شکل (۵-۲) مشاهده می‌کنیم این مدل‌های ابتدایی همراه با چیدمان فنر و دمپرها رسم شده‌اند.

¹ Maxwell

² Glass Compliance

³ Equilibrium Compliance

⁴ Relaxation



شکل (۶-۲) معرفی مدل‌های مکانیکی ابتدایی، الف) دمپر در مدل نیوتون ب) فنر و دمپر به صورت موازی در مدل کلونین ج) فنر و دمپر به صورت سری در مدل ماکسول [6]

□ مدل زینر^۱[6]

این مدل با اضافه کردن فنری به صورت سری با مدل کلونین یا به صورت موازی با مدل ماکسول به دست می‌آید. در این حالت شرایط $J_g > 0$ مد نظر است. در ادامه رابطه‌ی تنش و کرنش و همچنین ضریب خزش مدل زینر در معادله‌ی (۸-۲) آورده شده است.

$$\left[1 + a_1 \frac{d}{dt}\right] \sigma(t) = \left[M + b_1 \frac{d}{dt}\right] \epsilon(t) \quad (۸-۲)$$

$$J(t) = J_g + J_1 \left[1 - e^{-t/\tau_\epsilon}\right], J_g = \frac{a_1}{b_1} J_1 = \frac{1}{M} - \frac{a_1}{b_1}, \tau_\epsilon = \frac{b_1}{M}$$

از رابطه اخیر درمی‌یابیم که $0 < M < b_1 / a_1$ شرایطی است که J_1 را مثبت کرده و شرایط $0 < J_g < J_e < \infty$ نیز برقرار است.

□ مدل آنتی-زینر^۲[6]

^۱ Zener

^۲ Anti-Zener

این مدل نیز با اضافه کردن دمپری به صورت سری با مدل کلوین یا به صورت موازی با مدل ماکسول به دست می‌آید. در این حالت شرایط $J_g = \infty$ مدنظر است. در ادامه رابطه‌ی تنش و کرنش و همچنین ضریب خزش مدل آنتی-زینر در معادله‌ی (۹-۲) آورده شده‌است.

$$\left[1 + a_1 \frac{d}{dt}\right] \sigma(t) = \left[b_1 \frac{d}{dt} + b_2 \frac{d^2}{dt^2}\right] \epsilon(t) \quad (9-2)$$

$$J(t) = J_+ t + J_1 \left[1 - e^{-t/\tau_\epsilon}\right]$$

$$J_+ = \frac{1}{b_1} J_1 = \frac{a_1}{b_1} \frac{b_2}{b_1^2}, \quad \tau_\epsilon = \frac{b_2}{b_1}$$

از این رابطه درمی‌یابیم که برای مثبت شدن مقدار J_1 می‌بایست شرایط $0 < b_2/b_1 < a_1$ برقرار باشد. b_2 مربوط به جنس ماده می‌باشد.

□ مدل برگرز [6]^۱

اضافه کردن یک دمپر یا یک فنر به ترتیب به مدل زینر و آنتی-زینر مدل برگرز را به وجود می‌آورد. اگر این فنر یا دمپر را به صورت سری اضافه کنیم مدل خزش حاصل می‌شود. در نتیجه مدل برگرز مدلی است که از اتصال سری مدل ماکسول و کلوین بوجود می‌آید. در ادامه رابطه‌ی تنش و کرنش و همچنین مدل خزش در معادله‌ی (۱۰-۲) آورده شده‌است.

$$\left[1 + a_1 \frac{d}{dt} + a_2 \frac{d^2}{dt^2}\right] \sigma(t) = \left[b_1 \frac{d}{dt} + b_2 \frac{d^2}{dt^2}\right] \epsilon(t) \quad (10-2)$$

$$J(t) = J_g + J_+ t + J_1 \left(1 - e^{-t/\tau_\epsilon}\right)$$

برای به دست آوردن ثوابت با در نظر گرفتن تابع تنش به عنوان تابع پله از طرفین لاپلاس می‌گیریم که به صورت معادله (۱۱-۲) نوشته شده‌است.

¹ Burgers

$$\frac{1}{s} + a_1 + a_2 s = [b_1 s + b_2 s^2] E(s) \quad (11-2)$$

$$J(t) = -\frac{b_2}{b_1^2} + \frac{a_1}{b_1} + \frac{t}{b_1} + \left(\frac{a_2}{b_2} + \frac{b_2}{b_1^2} - \frac{a_1}{b_1} \right) e^{-\frac{b_1}{b_2} t}$$

$$, J_1 = \frac{a_1}{b_1} - \frac{a_2}{b_2} - \frac{b_2}{b_1^2} J_+ = \frac{1}{b_1} J_g = \frac{a_2}{b_2}$$

a_2 مربوط به جنس ماده می باشد.

□ مدل رابینسون [7]^۱

مدل ویسکوالاستیسیته ایزوتروپ عرضی که در ادامه معرفی می شود خود از مدل جامع تر خرابی و تغییر شکل که توسط رابینسون و همکارانش ارائه شده است، استخراج گردیده است. مدل حاضر توسعه یافته همان مدل اولیه است که تنش هیدرواستاتیک در پاسخ تغییر شکل وارد محاسبات شده است. مدل ویسکوالاستیک به صورت معادله ی (۱۲-۲) می باشد.

$$\dot{\epsilon}_{ij} = \dot{\epsilon}_{ij}^e + \dot{\epsilon}_{ij} \quad (12-2)$$

$$\frac{\dot{\epsilon}_{ij}}{\dot{\epsilon}_0} = \frac{3}{2} \Phi^{n-1} \frac{\Gamma_{ij}}{\sigma_0} \quad (13-2)$$

جایی که، $\dot{\epsilon}_{ij}^e$ نشان دهنده درایه های کرنش الاستیک، $\dot{\epsilon}_{ij}$ نرخ تغییر شکل کلی، $\dot{\epsilon}_{ij}$ نرخ خزش، σ_0 تنش مرجع، $\dot{\epsilon}_0$ و n پارامترهای مواد و Φ تابع پتانسیل اتلاف^۲ است. برای حالت ایزوتروپ عرضی، تابع Φ وابسته به تنش σ_{ij} ، تنش انحرافی s_{ij} و تنسور جهت گیری ماده D_{ij} می باشد. تابع پتانسیل اتلاف به صورت معادله ی (۱۴-۲) به دست می آید.

¹ Robinson

² Dissipation potential function

$$\Phi = \frac{1}{\sigma_0} \sqrt{3[J_2 - \xi(J - J_0^2) - (\xi - \eta)J_0^2 + (1/9)(\xi - 4\eta)I^2]} \quad (14-2)$$

که:

$$J_2 = \frac{1}{2} s_{ij} s_{ji}, J_0 = D_{ij} s_{ji}, I = \sigma_{ii}, I_0 = D_{ij} \sigma_{ji}, J = D_{ij} s_{jk} s_{ki} \quad (15-2)$$

و

$$\Gamma_{ij} = \frac{\partial \Phi}{\partial \sigma_{ij}} = s_{ji} - \xi(D_{ik} s_{kj} + s_{ik} D_{ij} - 2J_0 D_{ij}) - 2(\xi - \eta) \times J_0 \left(D_{ij} - \frac{1}{3} \mathbb{I}_{ij} \right) + \frac{2}{9} (\xi - 4\eta) I \mathbb{I}_{ij} \quad (16-2)$$

پارامترهای غیرایزوتروپ ξ ، η و ζ به صورت نامساوی‌های (۱۷-۲) تعریف می‌شوند.

$$\zeta - 4\eta \geq 0 \quad (17-2)$$

$$1 + 4\eta - 2\zeta \geq 0$$

$$4(\zeta - \eta) - 3\xi \geq 0$$

$$2(\zeta - \eta) - 3\xi \leq 0$$

این محدودیت‌های فیزیکی است که توسط رابینسون و همکارش مشخص شده‌اند.

شرایطی از مدل چند محوره را در نظر می‌گیریم که تحت برش و تنش یک‌جهته است. این حالت شامل برش

عرضی^۱ (TS)، برش طولی^۲ (LS)، نرمال عرضی^۳ (TN) و نرمال طولی^۴ (LN) است. شرایط خزش ثابت را در

این حالت تنش در نظر گرفته و بنابراین پاسخ تغییر شکل برای هر کدام به صورت معادله‌ی (۱۸-۲) است.

$$(TS) \frac{\dot{\gamma}_{23}/\sqrt{3}}{\dot{\epsilon}_{TS}} = \left(\frac{\sqrt{3}\tau}{\sigma_0} \right)^n, \dot{\epsilon}_{TS} = \dot{\epsilon}_0 \quad (18-2)$$

¹ Transverse Shear

² Longitudinal Shear

³ Transverse Normal

⁴ Longitudinal Normal

$$\begin{aligned} (LS) \frac{\dot{\gamma}_{23}/\sqrt{3}}{\dot{\epsilon}_{LS}} &= \left(\frac{\sqrt{3}\tau}{\sigma_0} \right)^n, \dot{\epsilon}_{LS} = \dot{\epsilon}_0 (1 - \xi)^{(n+1)/2} \\ (LS) \frac{\dot{\epsilon}_{22}}{\dot{\epsilon}_{LS}} &= \left(\frac{\sigma}{\sigma_0} \right)^n, \dot{\epsilon}_{TN} = \dot{\epsilon}_0 (1 - \eta)^{(n+1)/2} \\ (LN) \frac{\dot{\epsilon}_{11}}{\dot{\epsilon}_{LS}} &= \left(\frac{\sigma}{\sigma_0} \right)^n, \dot{\epsilon}_{LN} = \dot{\epsilon}_0 (1 - \zeta)^{(n+1)/2} \end{aligned}$$

□ مدل بیلی [8]^۱

یکی از مدل‌ها که در برای فازهای اولیه و ثانویه خزش متداول است مدل بیلی است که به صورت معادله (۲) - (۱۹) توسعه یافته است.

$$\dot{\epsilon}^{cr} = A \sigma^{-n} t^m \quad (۱۹-۲)$$

که پارامترهای A ، n و m ثابت‌های مواد وابسته به دما بوده و m به صورت کلی مستقل از تنش است. در حالی که n و m بی‌بعد می‌باشند. ضریب سخت شدگی کرنش خزش A ، دارای واحد بوده و شامل زمان و تنش است.

□ مدل فیندلی [9]^۲

فرم کلی مدل قانون فیندلی خزش به صورت رابطه‌ی (۲-۲۲) زیر است.

$$\epsilon = \epsilon_0 + m \left(t/t_0 \right)^n \quad (۲۰-۲)$$

¹ Bailey

² Findley

در جایی که ε کرنش خزش وابسته به زمان، ε_0 کرنش اولیه و m توابع تنش، n مستقل از تنش، t زمان و t_0 ثابتی است که به عنوان واحد در نظر گرفته می‌شود. بر طبق مدل فیندلی پارامترهای مدل خزش وابسته به تنش ε_0 و m توسط تابع هایپربولیک به صورت معادله‌ی (۲-۲۳) معرفی می‌گردد.

$$\varepsilon = \varepsilon_0 \sinh(\sigma/\sigma_0) + \varepsilon_t \left(t/t_0\right)^n \sinh(\sigma/\sigma_t) \quad (2-21)$$

در جایی که ε_0 ، ε_t ، σ_0 و σ_t ثابت‌های تجربی ماده هستند. برای مقادیر σ/σ_0 و σ/σ_t کمتر از واحد معادله بالا به صورت معادله‌ی (۲-۲۴) تبدیل می‌شود.

$$\varepsilon = \varepsilon_0 \sinh(\sigma/\sigma_0) + \varepsilon_t \left(t/t_0\right)^n (\sigma/\sigma_t) \quad (2-22)$$

$$\varepsilon = \sigma \left(\frac{1}{E_0} + \frac{t^n}{E_t} \right)$$

جایی که $E_0 = \sigma_0/\varepsilon_0$ مدول الاستیسیته اولیه مستقل از زمان بوده و $E_t = \sigma_t/\varepsilon_t$ مدولی است که رفتار وابسته به زمان دارد. حال مدول ویسکوالاستیک $E_v = \sigma/\varepsilon$ به صورت رابطه‌ی (۲-۲۵) دست می‌آید.

$$E_v = \frac{E_0 E_t}{E_t + E_0 t^n} \quad (2-23)$$

این معادلات برای هر دو حالت کرنش خزش کششی و کرنش خزش برشی مورد استفاده قرار می‌گیرند.

□ مدل شیپری [10]^۱

¹ Schapery

شیپری مدلی اساسی را بر اساس ترمودینامیک غیرقابل وارونپذیر برای پیش‌بینی کرنش خزش ویسکوالاستیک به صورت غیرخطی که در معادله‌ی (۲-۲۶) آمده است، ارائه داده است.

$$\varepsilon(t) = g_0 D_0 \sigma + g_1 \int_0^t \Delta D(\psi - \dot{\psi}) \frac{dg_2 \sigma}{dt} d\tau \quad (2-24)$$

جایی که D_0 و $\Delta D(\psi)$ ضرایب خزش ویسکوالاستیک خطی گذرا و لحظه‌ای بوده و Ψ زمان کاهش یافته‌ای است که به صورت معادله‌ی (۲-۲۷) تعریف می‌گردد.

$$\psi = \int_0^t \frac{d\dot{t}}{a_\sigma[\sigma(\dot{t})]} \quad (a_\sigma > 0) \quad (2-25)$$

$$\dot{\psi} = \psi(\tau) = \int_0^t \frac{d\dot{t}}{a_\sigma[\sigma(\dot{t})]} \quad (2-26)$$

ضرایب g_0 ، g_1 ، g_2 و a_σ همه پارامترهای غیرخطی وابسته به تنش هستند. پارامترهای g_0 ، g_1 ، g_2 مربوط به درجات بالاتر وابسته به انرژی آزاد گیبس به تنش اعمالی است. در حالی که a_σ نتیجه تأثیرات درجه بالاتر هر دوی تولید آنتروپی و انرژی آزاد است. این پارامترها از اصل‌های پایه انرژی نشأت گرفته و از منظر ترمودینامیک مهم هستند. ظهور این پارامترها در معادلات پایه بینشی برای چگونگی تأثیر آن‌ها در پاسخ ماده می‌دهد. ترم g_0 ، غیرخطی بودن را در ضریب الاستیک لحظه‌ای بر اساس تنش و دمای مختلف نشان می‌دهد. ضریب g_1 ، تفسیر مشابه‌ای دارد ولی بر ضریب خزش گذرا اثر دارد و g_2 ، تأثیرات غیرخطی بودن نرخ بارگذاری را نشان می‌دهد. پارامتر a_σ ، ضریب جابجایی زمان است که بستگی به تنش و دما دارد. مدل شیپری در دهه‌های اخیر در مدل‌سازی مواد مختلفی که رفتار ویسکوالاستیک غیرخطی دارند موفق بوده است.

۲-۵- مروری بر ادبیات موضوع

نیک و همکارانش [11] در سال ۲۰۰۸ به بررسی تعیین مشخصات ویسکوالاستیک میکرومکانیکی کامپوزیت‌های دارای الیاف پرداخته‌اند. یک الگوریتم محاسباتی کارآمد برای ارزیابی خواص ویسکوالاستیک کامپوزیت دارای الیاف پیشنهاد شده‌است. یک سلول واحد تکرار شونده^۱ دارای الیاف که ساختار کامپوزیت را نمایش می‌دهد. تحلیل المان محدود این سلول واحد تحت شش بارگذاری مشخص برای مطالعه و تعیین خواص ویسکوالاستیک کامپوزیت وابسته به زمان سلول واحد انجام شده‌است. الگوی میانگین حجمی اجرا شده برای تعیین پاسخ میانگین به عنوان تابعی از زمان برحسب تنش و کرنش استفاده می‌شود. این آنالیز داده‌ها برای توصیف کامپوزیت و تعیین پارامترهای ویسکوالاستیک و ضرایب به کار می‌برد. خواص ویسکوالاستیک منحصر به فرد مواد تشکیل دهنده همچنین کامپوزیت‌ها به منظور پیروی از سری پرونی تعریف می‌شود. برای تایید الگوریتم، مثال‌های مشخصی شامل می‌شوند. در این نمونه‌ها، مواد الیاف، الاستیک و ماتریس ویسکوالاستیک فرض می‌شود. سه نوع مختلف توزیع الیاف: شش گوشه‌ای^۲، مربع^۳ و الیاف متقاطع دو طرفه^۴، مورد استفاده قرار خواهد گرفت و هر توزیع برای درصد حجمی‌های مختلف الیاف (V_f / V) مورد بررسی قرار خواهد گرفت. دقت و صحت نتایج با شرایط شناخته‌شده مورد بررسی قرار گرفته‌است. هدف از این روش دقیق‌تر و کارآمد بودن قوانین ماده مرکب است. این روش میکرومکانیکی می‌تواند یک الگوریتم قدرتمند برای تعیین تعداد زیادی از خواص خطی و غیرخطی در خصوص مکانیک پیوسته و تحلیل و آنالیز ایجاد کند. مولیانا و ساوانت [12] در سال ۲۰۰۹ مدل میکرو مکانیکی برای پیش بینی واکنش کامپوزیت‌های پایه پلیمری تقویت شده با الیاف^۵ به زمان را ارائه کرده‌اند که اجزای تشکیل دهنده آن‌ها رفتارهای ویسکوالاستیک وابسته به دما را از خود نشان می‌دهند. کامپوزیت مورد مطالعه شامل

¹ RUC

² HEX

³ SQR

⁴ BCF

⁵ FRP

الیاف ارتوتروپیک و ماتریس همسانگرد است. خواص ماده ویسکوالاستیک برای الیاف و ماتریس، مجاز به تغییر با درجه حرارت می باشد. ریز ساختارهای کامپوزیتی (الیاف و ماتریس) به صورت الیاف مربعی توزیع شده به صورت متناوب در یک محیط ماتریسی، ایده آل سازی می شود. یک مدل سلول واحد که شامل چهار زیر سلول الیاف و ماتریس است تولید می شود تا واکنش های ویسکوالاستیک غیرخطی کامپوزیت ها به دست آید. الگوریتم یک پارچه سازی برای اتصال دو مدل متفاوت مواد ترمو - ویسکوالاستیک¹ در پایین ترین سطح (الیاف و ماتریس همگن) جهت دریافت پاسخ مناسب مواد در سطح ماکرو تشکیل شده است. این یک مدل میکرومکانیکی است که به عنوان خواص ماده ای اجزای تشکیل دهنده به درجه حرارت بستگی دارد. مدل های المان محدود سه بعدی برای چند نسبت حجمی مختلف استفاده شده است. نتایج آن مطابق با آزمون های تجربی بوده است. وانگ و همکارانش [13] در سال ۲۰۱۰ به شبیه سازی تولید خودکار و توزیع تصادفی الیاف در کامپوزیت های تقویت شده با الیاف پرداخته اند. الگوریتمی برای تولید خودکار دو بعدی المان حجمی نماینده^۲ از کامپوزیت های تقویت شده با الیاف بلند تک جهتی^۳ در این مقاله ارائه شده است. هم درصد حجمی الیاف و هم توزیع تصادفی الیاف در RVE در نظر گرفته می شوند. روشی که به عنوان آشفتگی محلی^۴ نام گذاری شد در این الگوریتم گنجانده شده است. بر اساس مدل تولید شده، تجزیه و تحلیل مزومکانیکی با استفاده از نرم افزار آباکوس با استفاده از روش المان محدود^۵ انجام می شود. در مرحله اول، اثر تصادفی توزیع الیاف در مدول عرضی بررسی شده است. ثانیاً، زیر برنامه کاربر برای تعریف مجدد متغیرها در یک نقطه ماده (USDFLD) در آباکوس برای شبیه سازی رفتار آسیب استفاده می شود. مجموعه ای از آزمایشات محاسباتی برای ارزیابی تأثیر اندازه مش بر بار نهایی کامپوزیت ها انجام می شود. نتایج به دست آمده ثابت می کند که الگوریتم قادر به ثبت ماهیت توزیع تصادفی از این مواد است و RVE تولید شده می تواند برای پیش بینی شروع آسیب و انتشار LFRC ها استفاده شود. اسدی و راقوان [14] در

¹ Thermo - viscoelastic

² rve

³ LFRCs

⁴ local disturbing

⁵ FEM

سال ۲۰۱۱ تاثیر خرابی وابسته به زمان را بر چند لایه‌های کامپوزیتی پایه پلیمری چندجهته مورد بررسی قرار دادند. مطابق نتایج به دست آمده، خزش در کامپوزیت به دلیل خواص ویسکوالاستیک لایه‌ها و همچنین خرابی وابسته به زمان رخ می‌دهد. پایانی‌کولا زوتسوس [15] در سال ۲۰۱۱ رفتار ویسکوالاستیک پلیمرها و کامپوزیت‌های پایه پلیمری را برای دو حالت خطی و غیر خطی مورد توجه قرار دادند. آن‌ها رفتار ویسکوالاستیک را در چهار بخش اصلی در نظر گرفتند. بخش اول مربوط به آزمایش‌های تجربی خزش و آسایش تنش است. بخش دوم مفهوم خطی بودن در ویسکوالاستیسیت را مورد بحث قرار می‌دهند و توضیحات کوتاهی در مورد اصل برهم نهی زمان - دما، زمان - تنش، زمان - دما - تنش را ارائه می‌کند. در بخش سوم انواع مختلف مدل‌های ویسکوالاستیک خطی از مدل‌های پایه ویسکوالاستیک تا مدل‌های تعمیم یافته مورد توجه قرار می‌گیرد. در نهایت بخش چهارم رفتار ویسکوالاستیک پلیمرها و کامپوزیت‌های زمینه پلیمری را مورد بررسی قرار داده و روش‌های مختلف پیش‌بینی در کنار کاربردهایشان ارائه می‌دهد. سا [16] در سال ۲۰۱۱ مروری کلی بر کارهای صورت گرفته در زمینه ویسکوالاستیک و رفتار خزشی وابسته به زمان پلیمرهای تقویت شده با الیاف پرداخته است. همچنین پلیمرهای تقویت شده پالترود شده^۱ در المان‌هایی با مقیاس‌های مختلف را نیز بررسی کرده‌است. پس از آن در این مقاله نتایج تحقیقات تجربی بر روی مواد پلیمری تقویت شده با الیاف شیشه پالترود شده که با پلی استر و الیاف شیشه در دو مقیاس مختلف ساخته شده‌اند ارائه شده‌است:

(۱) ورقه ورقه^۲

(۲) مشخصات در مقیاس کامل^۳

آزمون‌ها شامل: (۱) آزمون خزش خمشی^۴ دو نمونه با ضخامت ۸ میلی‌متر و در مقیاس کوچک با دهانه - ی ۱۶۰ میلی‌متر که تحت بارهای پایدار مربوط سطوح مختلف تنش با بازه‌ی ۲۰ تا ۸۰ درصد مقدار نهایی

¹ Pultruded glasss fiber reinforced polymers

² Laminate

³ Full - scale - profile

⁴ Flexural creep test

خود می‌باشد. و (۲) آزمون خزش خمشی در یک پروفیل I شکل در معرض 1/3 بار نهایی^۱ آن است. از جمله نتایج تجربی به دست آمده در این مطالعه این است که استحکام مشاهده شده در هر دو نوع مقیاس مواد نشان می‌دهد که این امکان وجود دارد که از نتایج کوپن مقیاس کوچک^۲ برای پیش‌بینی تغییر شکل خزش پروفیل‌های ساختاری استفاده کرد. منفرد و همکارانش [17] در سال ۲۰۱۲ یک فرمول تحلیلی جدید ارائه داده‌اند که به مطالعه‌ی خزش حالت پایدار^۳ در کامپوزیت‌های الیاف کوتاه با استفاده از روش متغیر پیچیده^۴ می‌پردازد. در این روش جدید هم الیاف و هم رزین تحت تنش‌ها و دماهای پایین قرار دارند. برای تجزیه و تحلیل الیاف تحت خزش از یک مدل تنش صفحه‌ای استفاده شده‌است. نوآوری مهمی که این روش تحلیلی دارد این است که تعیین نرخ جابجایی با شرایط مرزی مناسب در الیاف تحت خزش راحت بوده و همچنین استفاده از روش متغیر پیچیده در تحلیل خزش امکان‌پذیر است. این نکته قابل توجه است که این روش برای مطالعه‌ی رفتار خزشی در کامپوزیت‌های زمینه پلیمری با توجه به ظرفیت بالای خزش در آن مناسب است. علاوه بر این کاربرد قابل توجه دیگر این روش این است که می‌توان رفتار خزش یا پلاستیک کامپوزیت‌های پلیمری نانو تیوب کربن^۵ را مورد بررسی قرار دهد. در نهایت نتایج به دست آمده را روش تحلیلی حاضر (روش متغیر پیچیده) نشان می‌دهد که مطابقت خوبی با نتایج تجربی دارد. اسیون و همکارانش [18] در سال ۲۰۱۲ تعدادی آزمایش بر روی چند لایه‌های پلیمری تقویت شده با الیاف شیشه و فازهای تشکیل دهنده‌ی آن (الیاف و ماتریس) را تحت شرایط محیطی یکسان و سطح تنش‌های مختلف انجام دادند. پس از آن مدلی ارائه کردند که قادر به پیش‌بینی رفتار ویسکوالاستیک چند لایه‌های پلیمری تقویت شده با الیاف می‌باشد. نتایج نشان می‌دهد که این پیش‌بینی‌ها از اعتبار خوبی برخوردار است. مارک و همکارانش [19] در سال ۲۰۱۲ به یک روش عددی برای پیش‌بینی محل خرابی کامپوزیت‌های تقویت شده با الیاف تحت تراکم عرضی ترکیبی و بار محوری پرداخته‌اند. یک مدل محاسباتی مبتنی بر روش

¹ Ultimate load

² Small – scale - coupon

³ Steady – State - Creep

⁴ Complex variable method

⁵ Carbon Nano - Tube Polymer Composites

المان محدود برای تخمین قدرت یک لایه تقویت شده با الیاف در معرض ترکیبی از فشار عرضی و تنش محوری ارائه شده است. مکانیسم آسیب شامل شکست الیاف، گسیختگی الیاف / ماتریس و تغییر شکل پلاستیک ماتریس در مدل ارائه شده توسط معادلات تشکیل مناسب تولید می شود. شبیه سازی عددی پاسخ مکانیکی لایه تک جهته تحت بارگذاری دومحوره برای بدست آوردن محل خرابی استفاده می شود. پس از آن، مدل با یک راه حل تحلیلی و داده های تجربی مقایسه می شود. مشخص شد که محاسبات عددی با نتایج تجربی مطابقت بیشتری نسبت به پیش بینی های تحلیلی دارد. در سال ۲۰۱۳ چن و همکارانش [20] می دانستند خزش یکی از ویژگی های کامپوزیت های زمینه پلیمری است و می توان آن را به روش المان محدود تجزیه و تحلیل کرد اما تحت تاثیر عوامل مختلفی قرار دارد و به دست آوردن پارامترهای آن کار بسیار دشواری است. همچنین روش جابجایی پارامترها^۱ یکی از ساده ترین روش هایی است که می توان برای بدست آوردن خواص خزش با آزمون کمتر، مورد استفاده قرار بگیرد. آن ها در این پژوهش، روش تحقیق اساسی خزش در کامپوزیت های زمینه پلیمر را با روش المان محدود مورد مطالعه قرار دادند. تمرکز اصلی کار، این بود که چگونه پارامترهای مختلف ماده در تحلیل المان محدود خزش، نیازمند استفاده از روش جابجایی پارامترها می باشند. در نهایت یک ساختار واقعی لوله ی کامپوزیتی زمینه پلیمری با این روش بررسی شد. خطای نتیجه المان محدود و نتیجه تئوری حدود ۲ درصد به دست آمد. نتایج نشان می دهد که این روش منطقی بوده و از قابلیت خوبی برخوردار است. در سال ۲۰۱۴ سید موسوی و همکارانش [21] یک رویکرد میکرومکانیکی جدید برای بررسی شکست وابسته به زمان در کامپوزیت های زمینه پلیمری تک جهته را مورد توجه قرار دادند. مزیت اصلی این روش توانایی آن در ارائه حل تحلیلی برای رفتار غیر خطی کامپوزیت های تک جهته و پیش بینی پاسخ ماده به هر ترکیبی از بارگذاری برشی و نرمال است. شرایط شکست خزش نیز تابعی از شکست ماده ی زمینه ویسکوالاستیک در نظر گرفته شده است. ماده ی کامپوزینی که در این تحقیق استفاده شده است T300/934 بوده که برای استفاده در دمای بالا در صنعت هوافضا مناسب است. رفیعی و مظهری [22] در سال ۲۰۱۵ یک روش یک پارچه عمومی برای ارزیابی خزشی در

¹ Shift Factor Method

کامپوزیت‌های پایه پلیمری ارائه داده‌اند که با هر سازه‌ی کامپوزیتی قابل انطباق است. همچنین پدیده‌ی خزش غیرخطی با عمر بالای کامپوزیت‌ها بر اساس رفتار ویسکوالاستیک پلیمر شبیه‌سازی شده است. اندازه‌گیری شرایط کوتاه‌مدت تجربی بر روی رزین خالص برای شبیه‌سازی کامل خزش در چند لایه‌های کامپوزیتی، مورد نیاز است. روش مدل‌سازی در سطح یک لایه شروع می‌شود و نتایج به سطح چند لایه با تکیه بر روش گام به گام^۱ تعمیم داده می‌شود. در هر مرحله فرعی^۲ توزیع تنش/ کرنش در تطابق با خواص مکانیکی اصلاح شده به روز شده‌اند. انتخاب گام زمانی مناسب به منظور کاهش زمان اجرای مورد نیاز برای مدل‌سازی، توسعه داده شده است. نتایج حاصل از ارزیابی خزش نظری از مطابقت خوبی با داده‌های تجربی بلند مدت موجود برای صفحات کامپوزیتی برخوردار است. مطالعه موردی بر روی سازه‌های استوانه‌ای نیز انجام شده و نتایج مشاهده شده با داده‌های منتشرشده مقایسه شدند. نتیجه ارزیابی نشان داد که در موارد سازه‌های استوانه‌ای که تحت جهت‌گیری‌های مختلف الیاف و شعاع استوانه‌ای که در اثر خزش به وجود آمده است قرار گرفته‌اند، از اهمیت زیادی برخوردار است. در تحقیقات صورت گرفته توسط خلیلی و همکارانش [23] در سال ۲۰۱۵ در زمینه خزش کامپوزیت‌های پلیمری، رفتار خزشی کامپوزیت رزین فنولیک تقویت‌شده با پارچه بافته‌شده از الیاف بازالت به صورت تجربی مورد بررسی قرار گرفته است. در این پژوهش، دو نوع پارچه بافته شده از الیاف بازالت با بافت ساده^۳ و بافت اطلسی^۴ برای آزمایش مورد استفاده قرار گرفته است. برای بررسی اثر دما و تنش بر رفتار خزشی کامپوزیت تقویت شده با الیاف بازالت، نمونه‌های کامپوزیتی در سه دمای ۱۰۰ و ۱۵۰ و ۲۰۰ درجه سانتی گراد و همچنین دو بارگذاری مکانیکی با مقادیر ۳،۸ و ۶،۵ مگاپاسکال تحت آزمایش قرار گرفتند. مقایسه نتایج نشان داد نرخ کرنش در کامپوزیت‌های با بافت اطلسی نسبت به نمونه‌های با بافت ساده کاهش یافته است و این میزان کاهش تا حدود ۶۸ درصد در دمای ۲۰۰ درجه به دست آمد. همچنین پارامترهای دیگری نظیر عمر و مکانیسم گسیختگی کامپوزیت‌ها

¹ Incremental

² Sub - Step

³ Plane Weave

⁴ Strain Weave

مورد بررسی قرار گرفت. در همه نمونه‌ها در دمای ثابت با افزایش تنش، زمان گسیختگی و زمان تاخیر کاهش یافته و آهنگ کرنش خزشی افزایش می‌یابد. همچنین در تنش ثابت با افزایش دما زمان گسیختگی و تاخیر زمانی قطعه کاهش یافته و آهنگ افزایش می‌یابد. بررسی سطح شکست نمونه‌ها نشان داد که در پارچه بافته ساده، ابتدا لایه‌های بیرونی و سپس لایه‌های درونی گسیخته می‌شوند در حالی که در نمونه‌های بافت اطلسی تمامی لایه‌ها در یک زمان دچار گسیختگی شدند. تفاوت در مکانیسم انتقال تنش در نمونه‌ها عامل تفاوت در رفتار گسیختگی خزشی کامپوزیت است. اقدام و همکارانش [24] در سال ۲۰۱۵ به شبیه‌سازی آسیب در کامپوزیت‌های پایه فلزی تحت بار بارگذاری محوری با استفاده از مدل ناحیه چسبنده پرداخته‌اند. المان محدود، یک مدل میکرومکانیکی برای پیش‌بینی رفتار غیر الاستیک کامپوزیت‌های SiC / Ti در معرض بارگیری محوری با استفاده از یک عنصر RVE توسعه داده می‌شود. این مدل شامل تأثیر تنش‌های باقیمانده حرارتی فرایند تولید همراه با آسیب و نوع الیاف می‌باشد. از مدل ناحیه چسبنده برای در نظر گرفتن فصل مشترک بین الیاف و ماتریس استفاده می‌شود. معرفی معیار شکست منحصربه‌فرد برای زاویه‌های مختلف بارگذاری محوری، جدیدترین اصل این مقاله است. علاوه بر سطح مشترک آسیب، تغییر شکل پلاستیک ماتریس نیز به عنوان منبع دیگری از غیر خطی بودن در نظر گرفته می‌شود. شرایط مرزی مناسب بر RVE وضع شده است تا امکان استفاده همزمان بارمحوری و برشی عرضی و محوری به همراه تنش‌های باقیمانده حرارتی فراهم شود. نتایج مدل المان محدود ارائه شده در مقایسه با داده‌های تجربی برای پاسخ تنش - کرنش، شروع غیر خطی بودن، و مقاومت نهایی در زوایای بارگذاری محوری مختلف که تطابق خوبی با یکدیگر داشتند مقایسه می‌شوند. علاوه بر این مطالعات پارامتری به منظور بررسی اثرات تنش‌های حرارتی و درصد حجمی الیاف (FVF) در پاسخ مکانیکی ماده انجام شده است. انصاری و همکارانش [25] در سال ۲۰۱۵ به ارائه مدلی میکرومکانیکی بر پایه آنالیز ویسکوالاستیک کامپوزیت‌های تقویت شده با نانولوله کربنی در بارگذاری تک‌محوره و دوجوهی پرداخته‌اند. پاسخ ویسکوالاستیک نانو کامپوزیت‌های پلی‌آمید تقویت شده با نانولوله‌های کربنی (CNT ها) در معرض بارگذاری تک‌جهته و دوجته با استفاده از یک مدل میکرو مکانیکی براساس روش سلول - واحد مورد مطالعه قرار گرفته است. مدل میکرومکانیکی توسعه یافته

شده ساده و کارآمد است، و حالت‌های فرم بسته‌ای را برای پاسخ ویسکوالاستیک موثر نانوکامپوزیت‌ها را فراهم می‌کند. المان نماینده حجمی نانوکامپوزیت‌ها شامل سه مرحله شامل CNT های مداوم، ماتریس پلی آمید و فاز میانی است. حالت پخش CNT ها در ماتریس پلیمری، به صورت تصادفی است. نتایج ویسکوالاستیک و الاستیک به دست‌آمده مطابقت خوبی با روش‌های دیگر و داده‌های آزمایشگاهی دارد. سپس از این مدل برای مطالعه اثرات مواد میان‌فازی (الاستیک و ویسکوالاستیک) در رفتار خزش نانوکامپوزیت استفاده شده است. همچنین اثرات سطح استرس، شعاع CNT و فازمیانی در پاسخ ویسکوالاستیک نانوکامپوزیت تحت بارگذاری تک‌محوری و مساوی شامل عرضی / عرضی و طولی / عرضی مورد بررسی قرار می‌گیرد. سورزیا [26] در سال ۲۰۱۶ آزمون کششی تا هنگام شکست و آزمون‌های خزش و رهایی تنش را بر روی الیاف پلی‌پروپیلین انجام داد. پس از آن با برازش منحنی بر نتایج خزش و رهایی تنش حاصل از آزمایش‌ها و استفاده از کرنل کسری - نمایی^۱ مدل را ارائه کرده است. این مدل‌ها ساده‌تر از نمونه‌ها پیچیده‌تر هستند که از تلفیق فنر و دمپر حاصل شده‌اند علاوه بر این برای مطابقت داده‌های تجربی با تقریب خوبی مناسب هستند و هم‌زمان می‌توانند جهت تبدیل لاپلاس معکوس و به صورت تحلیلی مورد استفاده قرار گیرند، بنابراین پاسخ ویسکوالاستیک الیاف پلی‌پروپیلین را می‌توان به صورت مستقیم از طریق روش‌های تحلیلی مدل‌سازی کرد. اضافه کردن الیاف پلی‌پروپیلین به شدت باعث افزایش مقاومت کششی مواد کامپوزیتی با ماتریس بتن می‌شود. مدل تحلیلی پیشنهاد شده می‌تواند برای شبیه‌سازی رفتار مکانیکی مواد کامپوزیت با الیاف ویسکوالاستیک استفاده شود. درویزه و همکارانش [27] در سال ۲۰۱۶ از یک مدل میکرومکانیکی تحلیلی بر مبنای روش سلول واحد جهت مطالعه اثرات فاز میانی بر پاسخ ویسکوالاستیک غیرخطی کامپوزیت‌های زمینه پلیمری چندفازی استفاده کردند. به طور کلی المان حجمی نماینده کامپوزیت از سه فاز شامل الیاف هم‌راستا، زمینه پلیمری و فازمیانی بین الیاف و زمینه تشکیل شده و فرض اتصال کامل بین اجزای سازنده کامپوزیت منظور می‌شود در این پژوهش از مطالعه ساختاری ویسکوالاستیک شیپری برای مدل‌سازی رفتار ویسکوالاستیک غیرخطی زمینه استفاده شده است نتایج مدل

¹ Fraction – Exponential Kernel

میکرومکانیکی ارائه شده برای رفتار خزش پلیمر و کامپوزیت دوفازی تطابق بسیار خوبی را با داده‌های تجربی موجود نشان می‌دهد همچنین نتایج مدل حاضر برای خواص الاستیک کامپوزیت‌های سه‌فازی بسیار نزدیک به روش‌های عددی موجود است اثرات فازمیانی شامل مواد و ضخامت بر منحنی کرنش خزشی - بازیافت کامپوزیت سه‌فازی به طور متصل بررسی می‌شود نتایج نشان می‌دهد که خواص مواد تشکیل‌دهنده و ضخامت فازمیانی اثر بسیار مهمی بر پاسخ کرنش خزشی بازیافت کامپوزیت‌های سه‌فازی در بارگذاری عرضی دارد. همچنین طبق نتایج مدل‌سازی میکرومکانیکی مشخص شد که تاثیر فازمیانی بر رفتار ویسکوالاستیک غیرخطی کامپوزیت سه‌فازی تحت بارگذاری محوری ناچیز است به علاوه اثرات سطوح مختلف تنش و تغییرات درصد حجمی الیاف بر منحنی کرنش خزشی بازیافت کامپوزیت زمینه پلیمری سه‌فازی بررسی شده است. ژانگ و همکارانش [28] در سال ۲۰۱۸ برای توسعه یک روش جدید پیش‌بینی رفتار ویسکوالاستیک کامپوزیت‌های زمین پلیمری با توجه به رفتار الاستیک ماتریس از مدل ویسکوالاستیک ماتریس برای ساخت مدل‌های جدید برای کامپوزیت‌های تک‌جهته در راستای ۰ و ۹۰ درجه استفاده کرده‌اند پارامترهای ویسکوالاستیک برای هر دو مدل جدید استخراج شده اند و معادلات به‌دست‌آمده و فرمی همانند معادلات ساختاری ویسکوالاستیک ماتریس‌ها را دارند رفتارهای ویسکوالاستیک ماتریس و کامپوزیت‌های تک‌جهته مورد آزمایش قرار گرفتند و نتایج نشان داد که پارامترهای برازش شده رابطه نرمی خزشی به‌دست‌آمده از نتایج تجربی به مقادیر نظری در کامپوزیت‌های تک‌جهته خیلی نزدیک هستند و اعتبار مدل‌ها را اثبات می‌کنند در نهایت یک روش جدید برای پیش‌بینی خواص ویسکوالاستیک کامپوزیت‌های تک‌جهته براساس ویژگی‌های ویسکوالاستیک ماتریس و خواص الاستیک کامپوزیت‌های تک‌جهته به‌دست‌آمده است این روش یک مبنای تئوری برای مطالعات آینده بر روی رفتار ویسکوالاستیک چندلایه‌های کامپوزیتی را فراهم کرده است. احمدی و عطایی [1] در سال ۲۰۱۶ یک مدل میکرومکانیکی جهت بررسی رفتار خزشی و وابسته به زمان در مواد مرکب الیافی ارائه کردند در این پژوهش برای مدل‌سازی و استخراج رفتار کامپوزیت با استفاده از مدل میکرومکانیکی از مفهوم المان نماینده حجمی استفاده شده است با استفاده از روابط ساختاری مربوط به خزش در بارگذاری چند محوره، مدل المان محدود

برای مدل‌سازی خزش چندمحوره استخراج شده است. سپس روابط حاکم بر المان‌نماینده با در نظر گرفتن ترم‌های مربوط به خزش با استفاده از روش المان محدود گسسته‌سازی شده‌است و ماتریس‌های سفتی و نیرو متناسب با آن در مدل المان محدود استخراج شده‌است. با توجه به اینکه در بارگذاری‌های عمودی محوری و جانبی کامپوزیت الیافی ماهیت میدان جابجایی المان نماینده به صورت کرنش صفحه‌ای توسعه یافته قابل مدل‌سازی است لذا گسسته‌سازی معادلات برای حالت کرنش صفحه‌ای توسعه یافته ارائه شده‌است. شرایط مرزی المان‌نماینده متناسب با بارگذاری در مسئله خزش اعمال شده‌است و از روش صریح اوپلر برای حل معادلات در حوزه زمان استفاده شده‌است مدل میکرومکانیکی ارائه شده برای بررسی توزیع تنش در زمینه و الیاف و بررسی خزش مرحله‌ی پایدار کامپوزیت الیافی زمینه فلزی استفاده شده‌است و نقش خزش در تغییر توزیع تنش بر حسب زمان بررسی شده‌است سپس رفتار کامپوزیت در دیدگاه ماکرومکانیکی با استفاده از مدل میکرومکانیکی ارائه‌شده بر مبنای المان نماینده استخراج شده‌است خواص ماکرومکانیکی کامپوزیت برای بارگذاری‌های جانبی و محوری برای نسبت‌های حجمی متفاوت به‌دست آمده‌است. براردی و همکارانش [29] در سال ۲۰۱۷ برنامه آزمونی را ترتیب دادند تا رفتار ویسکوالاستیک کامپوزیت‌ها و فازهای آن را مورد توجه قرار دهند دراین راستا رفتار بلند کامپوزیت‌های زمینه پلیمری تک‌جهته با الیاف شیشه و فازهای آن (الیاف و ماتریس) را مورد بررسی قرار داده‌اند از اطلاعات تجربی که در مدت ۶ ماه به‌دست آمد یک مدل میکرومکانیکی برای پیش‌بینی رفتار چند لایه‌ها و الیاف پیشنهاد شده‌است در حالی که یک مدل غیرخطی پدیده‌شناختی^۱ که توسط نویسندگان ارائه شده‌است برای رزین مورد استفاده قرار می‌گیرد. خلیل و همکاران [30] در سال ۲۰۱۷ به چارچوب میکرومکانیکی با دقت بالا و محاسباتی با استفاده از مدل یک بعدی پرداخته‌اند. فان یه و همکارانش [31] در سال ۲۰۱۷ به ارائه یک کد پایتون ساده برای محاسبه خواص موثر المان نماینده حجمی دو بعدی و سه بعدی تحت یک شرایط مرزی متناوب پرداخته‌اند. که این کد می‌تواند خواص مکانیکی کامپوزیت با سازه متناوب را در دو بعد یا سه بعد محاسبه کند. روش محاسبه برپایه تئوری همگن‌سازی تقریبی (AHT) است. به اصطلاح ساده، کد پایتون پایه ممکن است برای محاسبه

¹ Phenomenologicals

خواص موثر ساختار میکرو پیچیده تر گسترش یابد. اگر چه این کد بستر مناسبی روی بهینه سازی برای مواد و طراحی کامپوزیت ژئومتریک تهیه می کند. کاربر ممکن است با الگوریتم های مختلف آزمایش کند و گستره ی وسیعی از مشکلات را رفع کند. برای اثبات اثربخشی و قابلیت اطمینان کد، یک مورد سه بعدی برای روشن کردن کد بکار گرفته شده است. در آخر نتایج عددی بدست آورده شده توسط کد مطابقت خوبی با نتایج آزمایشگاهی و تجربی در دسترس داشته است. ژیرلی و همکاران [32] در سال ۲۰۱۹ به مدل سازی عددی و اعتبارسنجی آسیب حرارتی - مکانیکی در کامپوزیت اپوکسی DER353 / شیشه boronilicate با تغییر شکل نرخ کرنشی بالا پرداخته اند. نادری و همکاران [33] در سال ۲۰۱۹ به آنالیز میکرو مکانیکی، مکانیزم آسیب تحت تنش لایه های کامپوزیتی باریک با زاویه ۹۰ و ۰ درجه پرداخته اند. سید علی خانی و همکاران [34] در سال ۲۰۲۰ به ارائه مدل میکرو مکانیکی و ساختار دینامیکی جدید برای پیش بینی رفتار مکانیکی وابسته به نرخ کرنش کامپوزیت های چند لایه شیشه / اپوکسی پرداخته اند. سانتوش و همکارانش [35] در سال ۲۰۲۰ به بررسی تأثیر تخریب بر رفتار غیر خطی و وابسته به زمان کامپوزیت های پایه سرامیکی پرداخته اند. تأثیر تخریب در پاسخ به تنش - کرنش غیر خطی و پاسخ کرنش وابسته به زمان تحت تنش ثابت بررسی شده است. مواد مورد استفاده در این مقاله کامپوزیت بافته شده از الیاف CG-Nicalon و ماتریس SiNC است. که در آن از یک مدل میکرو مکانیکی استفاده شده است. تخمین های عددی تغییر شکل و خرابی در مقایسه با اندازه گیری های تجربی نشان داده شده است. همچنین نشان داده شد که تخریب با افزایش تخریب در مواد، ویژگی های ضخامت - ضخامت را بیشتر تحت تأثیر قرار می دهد و مقاومت گسیختگی خزش به طور قابل توجهی با افزایش تخریب در مواد تخریب می شود. الطاهر و همکاران [36] در سال ۲۰۲۰ به بررسی مدل سازی میکرو مکانیکی آسیب در نانو کامپوزیت های الاستیک - پلاستیک با استفاده از سلول واحد نماینده المان حجمی و مدل ناحیه چسبنده پرداخته اند. هانگ بو هانگ [37] در سال ۲۰۲۰ به پیش بینی میکرو مکانیکی رفتار الاستیک - پلاستیک، الیاف کوتاه یا کامپوزیت تقویت شده با ذره پرداخته اند. گودارا و همکاران [38] در سال ۲۰۲۰ به مطالعه روش میکرو مکانیکی برای آنالیز مواد کامپوزیتی پرداخته اند. شعافی و همکاران [39] در سال ۲۰۲۰ به بررسی مدل آسیب بر پایه میکرو مکانیکی

برای آنالیز رفتار فشاری لایه‌های کامپوزیتی پرداخته‌اند. موناتایرسکیز و همکاران [40] در سال ۲۰۲۰ به ارائه مدل میکرومکانیکی کامپوزیت‌های پلیمری MXenee پرداخته‌اند. فرانسسکا و همکاران [41] در سال ۲۰۲۰ به ارائه مدلی میکرومکانیکی و پاسخ سازنده کامپوزیت‌های FRCM پرداخته‌اند. بین بین و همکاران [42] در سال ۲۰۲۰ به آنالیز میکرومکانیکی رفتار خزشی ناشی از انتشار فصل مشترک در کامپوزیت‌های پایه فلزی تقویت شده با الیاف تک‌جهته تحت شرایط بارگذاری عمومی پرداخته‌اند.

۲-۶- نتیجه‌گیری

در این فصل به انواع مقیاس‌ها در کامپوزیت‌ها و مدل ویسکوالاستیک و انواع مدل‌های خزش و مقالات مرور شده در این حوزه پرداخته شده‌است.

فصل ۳:

روش تحقیق

۳-۱- مقدمه

در این فصل با استفاده از کد پایتون که در پیوست یک به طور کامل توضیح داده شده است، به بسط مدل قانون توان و اعمال آن در نرم افزار اجزا محدود آباکوس می پردازیم. نوع ماده استفاده شده در این گزارش کامپوزیت با زمینه پلیمری است که امروزه استفاده از این مواد در کاربردهای سازه ای به دلیل وزن کم و دوام بالا بسیار چشم گیر شده است. با این حال همچنان مدل سازی عددی این مواد چالش های متعددی دارد. دلیل آن نیز به علت طبیعت ناهمسانگردی بالای خزش است. همچنین خرابی مواد باعث علاقه مند شدن طراحان در استفاده از کامپوزیت های دارای زمینه پلیمری است.

هنگامی که کامپوزیت های دارای زمینه پلیمری در محدوده خزش خود عمل کنند دچار تغییر شکل وابسته به زمان شده و در نهایت به خرابی می رسند. بیشتر بار در این مواد توسط فایبرهای مستحکم تحمل شده و خزش اگر وجود داشته باشد بسیار کم است.

۳-۲- بسط مدل قانون توان

یکی از مدل ها که در برای فازهای اولیه و ثانویه خزش متداول است مدل توانی است که به صورت معادله (۳-۱) توسعه یافته است.

$$\dot{\epsilon}^{cr} = A \sigma^{-n} t^m \quad (3-1)$$

که پارامترهای A ، n و m ثابت های مواد وابسته به دما بوده و m به صورت کلی مستقل از تنش است. در حالی که n و m بی بعد می باشند. ضریب سخت شدگی کرنش خزش A ، دارای واحد بوده و شامل زمان و تنش است. با انتگرال گیری معادله بالا نسبت به زمان معادله ی (۳-۲) خواهیم داشت.

$$\varepsilon = A\sigma \frac{t^{m+1}}{m+1} + \varepsilon_0 \quad (2-3)$$

و همچنین با جایگذاری قانون هوک در معادله (۲-۳) به معادله نرمی خزشی (۳-۳) می‌رسیم.

$$\frac{1}{E} = A \frac{t^{m+1}}{m+1} + \frac{1}{E_0} \quad (3-3)$$

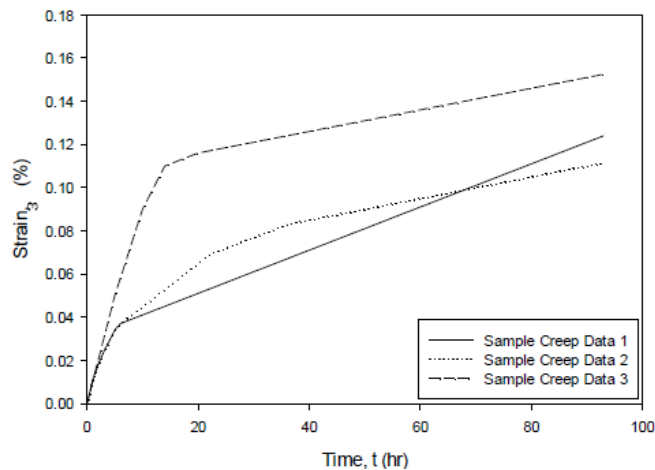
مدل توانی به صورت معادله‌ی (۴-۳) نیز ارائه شده است.

$$\dot{\varepsilon}_{cr} = A' \sigma^{n'} t^{m'} \quad (4-3)$$

جایی که A' ، n' و m' به صورت ثابت‌های وابسته به دما باقی می‌مانند و A' دارای واحد

$MPa-n'hr-m'-1$ است. در این حالت می‌بایست مقدار A' بزرگتر از صفر باشد. این مدل‌ها برای

مواردی مناسب است که در سابقه خزش، فاز اولیه و ثانویه خزش قالب باشد.



شکل (۳-۱) مثالی از خزشی که فاز اولیه و ثانویه غالب است [8]

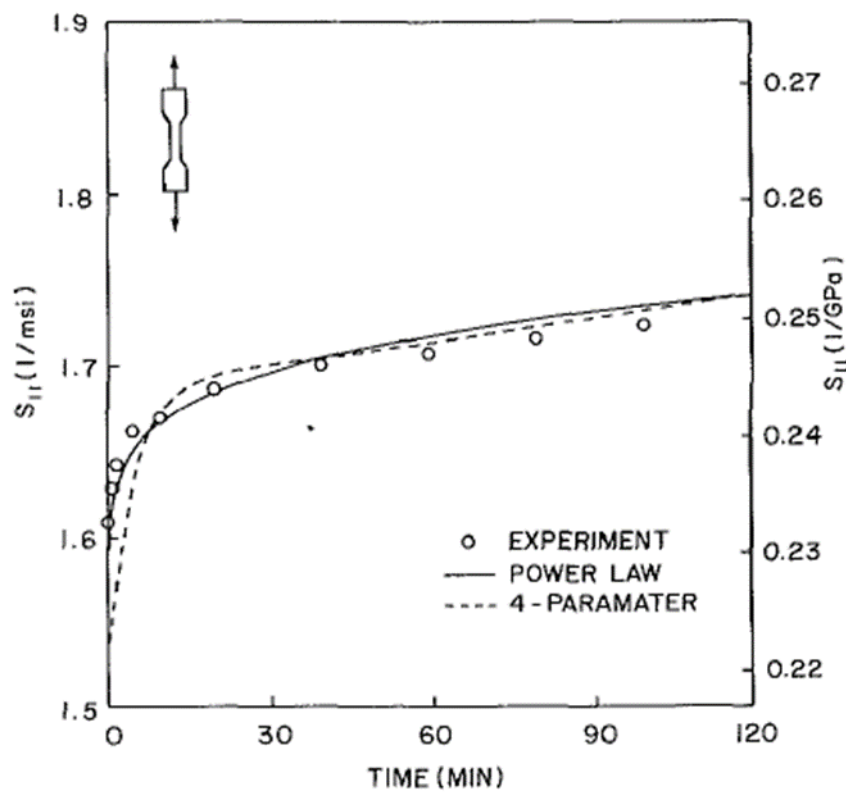
برای ادامه کار و برای محاسبات از مدل پاور - لا استفاده می‌شود.

۳-۳- استفاده از مدل پاور-لا در آباکوس

برای استفاده از مدل قانون توانی (۱-۳) در نرم افزار آباکوس نیاز به ضرایب ثابت A و m است. ضرایب A و m یک نمونه اپوکسی با استفاده از برازش منحنی معادله (۳-۳) روی نتایج آزمایشگاهی مرجع [43] (شکل (۲-۳)) بدست آمد که در جدول (۱-۳) ارائه شده است.

جدول (۱-۳) ثابت‌های به دست آمده از برازش منحنی معادله پاور-لا

A	M
1.10E-06	-0.8218



شکل (۲-۳) داده‌های آزمایشگاهی مورد استفاده جهت محاسبه ضرایب قانون توانی [43]

همچنین خواص الاستیک رزین و الیاف کربن به ترتیب طبق جدول (۲-۳) و جدول (۳-۳) در نظر گرفته شده است [43].

جدول (۲-۳) خواص الاستیک رزین

$E_m(\text{Mpa})$	ν_{12m}
4510	0.311

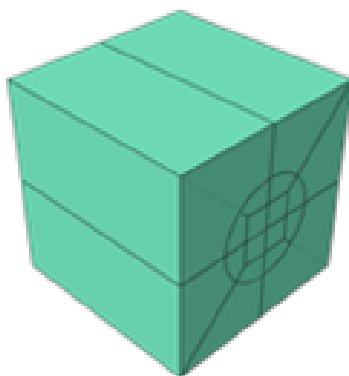
جدول (۳-۳) خواص الاستیک الیاف کربن

$E_1(\text{Mpa})$	$E_2(\text{Mpa})$	$E_3(\text{Mpa})$	ν_{12}	ν_{13}	ν_{23}	$G_{12}(\text{Mpa})$	$G_{13}(\text{Mpa})$	$G_{23}(\text{Mpa})$
۲۰۲۸۲۰	۲۵۳۰۰	۲۵۳۰۰	۰,۴۴۳	۰,۴۴۳	۰,۰	۴۴۱۲۰	۴۴۱۲۰	۴۴۱۲۰

۳-۴- مدل سازی سلول واحد

در تحلیل میکرومکانیک برای تحلیل خواص ماده بر حسب اجزای تشکیل دهنده آن ماده از المان حجمی نماینده استفاده می شود. هدف از به کار گیری المان نماینده برقراری رابطه ای بین خواص ماده از دیدگاه ماکرو به ویژگی های ریز ساختار و خواص اجزای تشکیل دهنده آن می باشد. [44] المان حجمی نمونه در مسائلی که مدل ماکرو دارای خواص المان های مشابه و پیرو دیک بوده استفاده می شود. در کامپوزیت ها مدل میکرومکانیک شامل ماتریس و فایبر می باشد که فایبر با هندسه دایروی درون ماتریس قرار دارد. در تحلیل های میکرومکانیکی، مدل سازی کل سازه یا ماده ممکن است، ولی از لحاظ مدل سازی پیچیده و از لحاظ محاسباتی بسیار زمان بر خواهد بود بنابراین از المان نماینده حجمی استفاده می شود. هر چند که توزیع الیاف در سطح مقطع کامپوزیت الیافی تصادفی است، ولی برای مدل سازی میکرومکانیکی معمولاً فرض می شود که الیاف دارای قطر یکسان بوده و به صورت منظم در آرایش مربعی در سطح مقطع کامپوزیت توزیع شده است. معمولاً کوچکترین جزء تکرار شونده از سطح مقطع کامپوزیت به عنوان المان حجمی نماینده

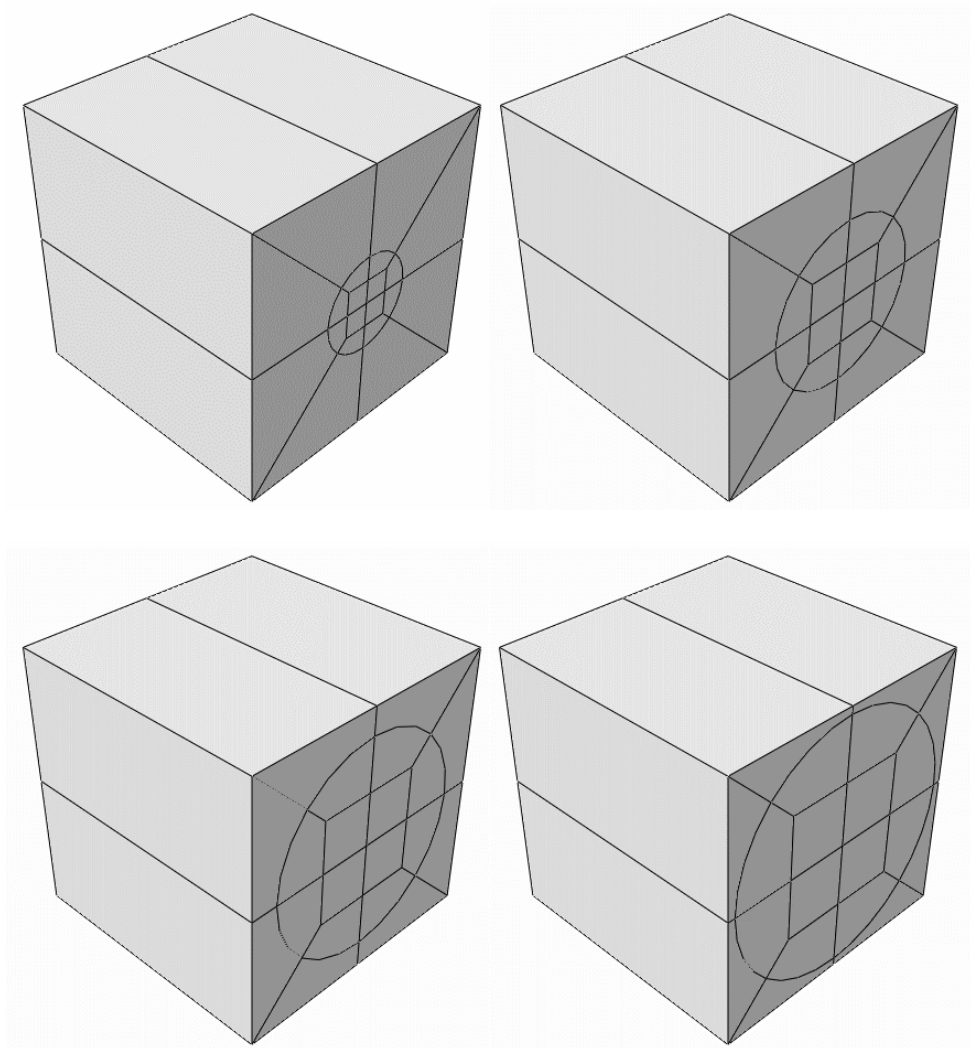
کامپوزیت در نظر گرفته می‌شود و در تجزیه و تحلیل میکرومکانیکی فرض می‌شود که می‌توان رفتار ماده مرکب را با مدل‌سازی صحیح المان نماینده انجام داد. المان نماینده انتخاب شده برای کامپوزیت الیاف بلند در "شکل ۱" نشان داده شده است.



شکل (۳-۳) المان حجمی نماینده انتخاب شده

انتخاب المان نماینده به نوع بار اعمالی بستگی دارد و معمولاً برای کاهش زمان انجام محاسبات کوچکترین قسمت ممکن که بتوان شرایط مرزی مربوط به بارگذاری را به صورت صحیح بر آن اعمال کرد به عنوان المان حجمی نماینده انتخاب می‌شود. با اعمال درست شرایط مرزی و شرایط بارگذاری می‌توان خواص مکانیکی و رفتار کامپوزیت‌ها را پیش‌بینی کرد. در این پایان‌نامه فرض می‌شود که الیاف در جهت X بوده و محورهای Y و Z جهت‌های عمود بر الیاف می‌باشند. صفحه سطح مقطع ماده $Y-Z$ ، در جهت X همگن می‌باشد.

مدل اعمالی در نرم افزار آباکوس یک مدل $1 \times 1 \times 1$ می‌باشد که فایبر با درصد حجمی‌های مختلف ۰/۱ و ۰/۳ و ۰/۵ و ۰/۷ در حالت تک فایبر در مرکز ماتریس قرار گرفته‌است. که به ترتیب در شکل (۳-۴) نشان داده شده‌است.



شکل (۳-۴) سلول واحد با درصد حجمی‌های به ترتیب ۰/۱ و ۰/۳ و ۰/۵ و ۰/۷

در حالت تک‌فایبر دارای فازمییانی همانند حالت تک‌فایبر یک سلول $1 \times 1 \times 1$ می‌باشد که فایبر با درصد حجمی ۰/۵ درون ماتریس قرار گرفته‌است. قسمت فازمییانی به ترتیب در ۵ مدول الاستیسته مختلف $Em_{0.1}$ و $Em_{0.3}$ و $Em_{0.5}$ و $Em_{0.6}$ و در خود Em در ضخامت فازمییانی ۰/۰۸ انجام شده‌است. [45]

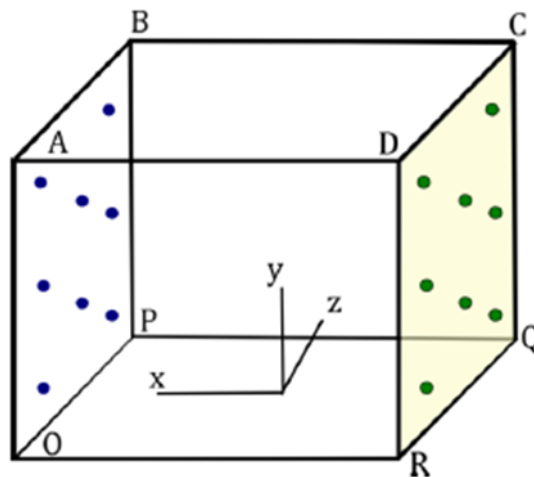
در حالت تولید تصادفی الیاف، با تولید تصادفی چهار فایبر درون ماتریس به ترتیب در سه نسبت حجمی ۰/۳ و ۰/۵ و ۰/۵۵ صورت گرفته‌است.

و همچنین مقدار بار وارده بر سلول واحد ۱ مگاپاسکال می‌باشد و تحت شرایط مرزی متناوب قرار گرفته‌است.

۳-۵- شرایط مرزی متناوب

در استفاده از شرایط مرزی متناوب مشطوری انجام شده که دقیقاً با یک شکاف کمی نودهای متناظر در دو صفحه روبرو به همدیگر باشند که برای رسیدن به این موضوع یک پارتیشن بندی منظمی انجام شده که این باعث می‌شود که دقیقاً نودهای متناظر روبروی هم دیگر قرار گیرند و سپس با الگوریتم سرچ، نودهای متناظر را یافته و شرایط مرزی را طبق معادله (۳-۵) اعمال می‌کند.

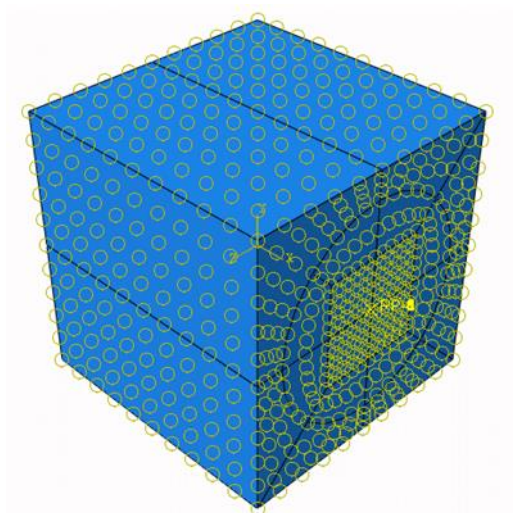
$$u_i^{j+} - u_i^{j-} = \bar{\varepsilon}_{ik} \Delta x_k^j \quad (۳-۵)$$



شکل (۳-۵) مثالی از نودهای متناوب و غیر متناوب روی صفحه‌های x مثبت و منفی

استفاده از شرایط مرزی متناوب در به حداقل رساندن اثرات لبه کمک می‌کند. برای به دست آوردن برش خالص نیاز است که نودها در دو صفحه و به صورت متناظر با یکدیگر کوپل شده و رفتاری مشابه داشته باشند. این مهم توسط قید Equation در نرم افزار المان محدود آباکوس حاصل می‌گردد مساله‌ای که بسیار حایز اهمیت می‌باشد قرار گرفتن نودها در دو صفحه متناظر به صورت کاملاً متقارن است شرایط

مرزی متناوب علاوه بر تغییر شکل یکسان از جدایش و هم‌پوشانی سلول‌های همسایه جلوگیری می‌کند.



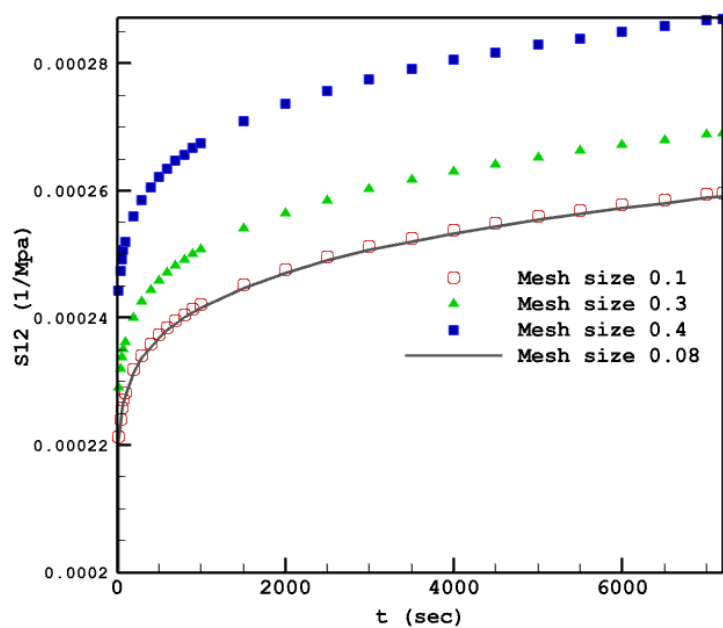
شکل (۳-۶) شرایط مرزی اعمالی بر سلول واحد و کیفیت کوپل شدن نودهای دو صفحه‌ی مجاور توسط کد پایتون

برای اینکه دقیقاً به صورت نود به نود عمل کند ابتدا باید مش در وجه‌های مقابل بسیار شبیه هم باشند و سپس با استفاده از اسکریپت نویسی نودهای مقابل انتخاب شود و قید equation به آن‌ها اعمال شود. برای مثال برای اعمال معادله $U_1 + 2U_3$ در ستون coefficient به ترتیب اعداد ۱ و ۲ اعمال شود و در ستون set name هم ست‌های معرف هر یک از دو ناحیه را انتخاب شود و در ستون dof هم به ترتیب اعداد ۱ و ۳ گذاشته شود.

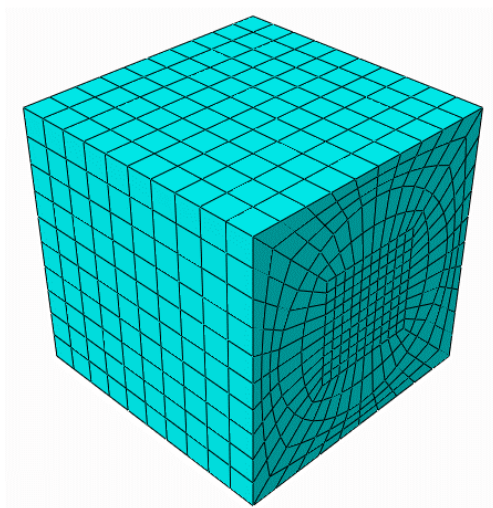
۳-۶- شبکه بندی مدل

برای به‌دست آوردن شبکه بندی مناسب با استفاده از قیدهای بخش‌بندی در آباکوس مدل را به قسمت‌هایی با سایز ۰,۱ تقسیم می‌کنیم. شکل (۳-۸) تعداد المان در مدل را نشان می‌دهد. در بررسی شبکه‌بندی جنس فایبر از جدول (۳-۴) و ماتریس از جدول‌های (۳-۲) و (۳-۳) انتخاب شده و تحت بار خزشی ۱

مگاپاسکال و در مدت ۷۲۰۰ ثانیه تحلیل شده‌است. و همچنین نمودار حساسیت به مش در سائز المان‌های ۰/۸ و ۰/۱ و ۰/۳ و ۰/۴ رسم شده‌است همانطور که در شکل (۷-۳) مشاهده می‌شود نتایج اختلاف کمی با یکدیگر دارند و برای انجام مدل‌سازی‌ها از سائز المان ۰,۱ استفاده شده‌است.



شکل (۷-۳) نمودار حساسیت به مش



شکل (۳-۸) شبکه بندی سلول واحد با سایز ۰/۱

تمام مدل سازی های انجام شده به وسیله کد پایتون صورت گرفته است. تمام کد پایتون های استفاده شده در پیوست ۱ آورده شده است و همچنین کد استفاده شده در زیر توضیح داده شده است.

۱. مقدار دهی پارامترها

۲. ایجاد درصد حجمی های مورد نظر

۳. ایجاد model 1

۴. محاسبه شعاع الیاف با استفاده از معادله
$$R = \sqrt{\frac{V_f \times L_x \times L_y}{\pi \times N}}$$

۵. ایجاد قسمت پارت

۶. اعمال خواص ماده برای ماتریس و فایبر و اختصاص دادن هر خواص را به پارت مورد نظر

۷. اسمبلی کردن فایبر و ماتریس

۸. ایجاد استپ ویسکوز

۹. ایجاد set های reference point

۱۰. ایجاد نودهای مربوط به شرایط مرزی متناوب در سطح (به جز لبه ها و گوشه ها)

۱۱. ایجاد Set های برای نودهای لبه (به جز گوشه ها)

۱۲. ایجاد Set هایی برای نودهای گوشه

۱۳. اعمال چهار شرایط مرزی

۱۴. اختصاص BC_1 را برای E_{11} و BC_2 را برای E_{22} و BC_3 را برای E_{33} و BC_4 را برای E_{12} و BC_5 را برای

E_{13} و BC_6 را برای E_{23}

۱۵. در قسمت شرایط مرزی مقدار بار را ۱ مگاپاسکال و زمان نیز ۷۲۰۰ ثانیه در نظر گرفته شده است.

۱۶. ایجاد JOB

۱۷. دریافت دو انرژی Strain energy و Creep dissipation energy

۱۸. با توجه به معادله انرژی کل $2U = \sigma^2 / E$ و با استفاده از دو انرژی به دست آورده انرژی کل را محاسبه

می کند و از آنجا که تنش ما ۱ می باشد $2U = 1/E$ و می دانیم که نرمی خزشی معکوس مدول است

پس انرژی کل به دست آمده همان کرنش می باشد.

برای حالت فازمیانی در قسمت پارت، یک فازمیانی به ضخامت ۰,۰۸ ایجاد شده و در قسمت خواص، سه

مدول مختلف $Em \ 0/1$ و $Em \ 0/3$ و $Em \ 0/0.5$ و در خود Em برای فازمیانی در نظر گرفته شده و

محاسبات انجام شده است و بقیه روند کار مانند توضیح قست قبل می باشد. همچنین برای کد تولید رندم

به همین منوال است و فقط نحوه تولید الیاف رندم در زیر توضیح داده شده است.

۳-۷- تولید تصادفی الیاف

برای تولید تصادفی الیاف در ماتریس از یک کد پایتون استفاده شده که در آن کد دارای ورودی های زیر

می باشد:

➤ تعداد الیاف با مرکز فایبر

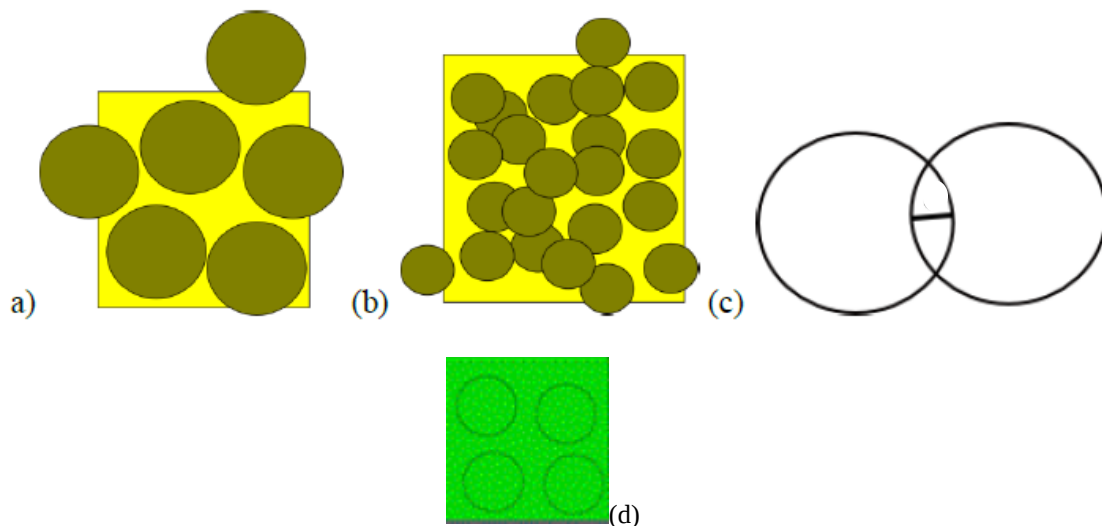
➤ کسر حجمی الیاف

➤ پارامترهای ساختاری الیاف و ماتریس

➤ اندازه شبکه بندی الیاف و ماتریس

که در این کد تعداد الیاف ۴ می باشد که با هربار ران کردن مختصات مرکز الیاف به دلیل تصادفی الیاف تغییر می کند. و همچنین درصد حجمی های استفاده شده در این کد ۰/۳ و ۰/۵ و ۰/۵۵ است که در ۴ شرایط مرزی مختلف انجام شده است. و همچنین خواص ماتریس و الیاف همانند جدول (۳-۲) و جدول (۳-۳) می باشد و اندازه شبکه بندی الیاف و ماتریس ۰/۰۵ در نظر گرفته شده است.

در شکل (۳-۹) نمونه ای از قرار گرفتن تصادفی الیاف در ماتریس مشاهده می شود. که RVE (a) با الیاف مجاز به خروج از مرز سلول واحد RVE (b) با الیاف هایی که از مرز سلول واحد عبور کرده و خارج می شود (c) RVE تعریف فاصله تقاطع (d) RVE با الیاف هایی که کامل درون سلول واحد قرار گرفته اند.



شکل (۳-۹) نحوه قرار گرفتن تصادفی الیاف در RVE.

نحوه کار اسکریپت به صورت زیر می باشد:

در ابتدا با توجه به طول های L_x و L_y و تعداد الیاف و با توجه به معادله (۳-۶) شعاع الیاف به دست آمده است.

$$R = \sqrt{\frac{V_f \times Lx \times Ly}{\pi \times N}} \quad (6-3)$$

Lx فاصله لبه از x

Ly فاصله لبه از y

V_f درصد حجمی الیاف

N تعداد الیاف

سپس فاصله بین دو نود با توجه به معادله (۷-۳) محاسبه شده است.

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (7-3)$$

در ادامه با استفاده از معادله (۸-۳) بردار واحد محاسبه شده است.

$$Z = \left[\frac{x_1 - x_2}{d}, \frac{y_1 - y_2}{d} \right] \quad (8-3)$$

که r همان فاصله بین دو نود می باشد. در ادامه کد بررسی می کند که آیا فایبر داخل RVE قرار دارد یا نه که با توجه به معادله (۹-۳) بررسی شده است.

$$\begin{cases} X > \frac{Lx}{2} - R \\ X < -\frac{Lx}{2} - R \end{cases} \quad (9-3)$$

$$\begin{cases} Y > \frac{Ly}{2} - R \\ Y < -\frac{Ly}{2} - R \end{cases}$$

X و Y مختصات الیاف می باشد.

در ادامه کد به تعیین مکان های الیاف با توجه به معادله (۱۰-۳) می پردازد.

$$C = \left[\left[\left(-\frac{Lx}{2} + R \times (1 + tol), \frac{Lx}{2} - R \times (1 + tol) \right), \right. \right. \\ \left. \left. \left(-\frac{Ly}{2} + R \times (1 + tol), \frac{Ly}{2} - R \times (1 + tol) \right) \right] \right] \quad (10-3)$$

tol تolerانس اولیه می باشد.

و بر اساس معادله (3-11) مختصات الیاف رندم به دست می آید.

$$X = -\frac{Lx}{2} + R \times (1 + tol), \frac{Lx}{2} - R \times (1 + tol) \\ Y = -\frac{Ly}{2} + R \times (1 + tol), \frac{Ly}{2} - R \times (1 + tol) \quad (11-3)$$

و همچنین فاصله بین الیاف ها را طوری می سنجد که با یکدیگر برخوردی نداشته باشند.

در این کد نحوه قرار گرفتن الیاف به این صورت است که هر چهار الیاف در داخل سلول واحد قرار دارند و با

یکدیگر تداخل ندارند. نتایج به دست آمده در فصل بعد آورده شده است.

۳-۸- جمع بندی

در این فصل با استفاده از یک کد پایتون به بسط مدل توان و اعمال آن و نحوه مدل سازی در نرم افزار اجزا محدود آباکوس پرداخته شد.

فصل ۴:

نتایج و تفسیر آنها

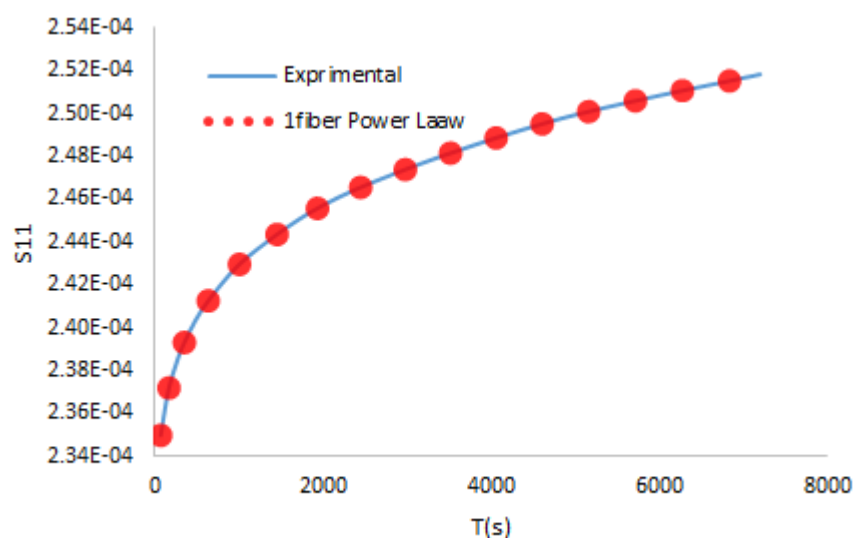
۴-۱- مقدمه

در این فصل ابتدا به صحّه گذاری مدل پرداخته می‌شود و در ادامه به مقایسه نتایج به‌دست آمده در سه مدل تک فایبر و تک فایبر دارای فاز میانی و حالت تولید تصادفی پرداخته می‌شود.

۴-۲- صحّه گذاری مدل

در این قسمت نتایج به دست آمده، با نتایج تجربی [43] مقایسه شده‌است. در مدل قانون توان تنش ثابت در نظر گرفته می‌شود. با استفاده از برازش غیرخطی دو پارامتر دیگر مدل شامل A و m ، نسبت به زمان تخمین زده شده‌است. نتایج ضرایب یافت شده از این روش در جدول (۳-۱) مشاهده می‌شود.

در ادامه همانطور که در بخش قبل توضیح داده شد با استفاده از الگوریتم ارائه شده برای مدل قانون توان با اعمال کد پایتون به حل یک مدلی تک‌المانه در جهت یک پرداخته شده‌است. نوع مدل و شرایط مرزی آن به صورتی در نظر گرفته شده است که متناسب با شرایط یک‌بعدی بوده که تنها تنش نرمال در جهت یک بر آن اثر کند. این تحلیل با در نظر گرفتن ضریب پواسون $\nu = 0.31$ و در مدت زمان ۲ ساعت صورت گرفته‌است. پس از اتمال تحلیل نتایج بدست آمده با نتایج [43] مقایسه شده‌است که در شکل (۴-۱) آورده شده‌است. که می‌توان تطابق خوبی بین آن‌ها مشاهده کرد.



شکل (۴-۱) مقایسه مدل آزمایشگاهی مرجع [43] با مدل تک‌فایبر

۴-۳- تعیین خواصّ مدل

همانطور که گفته شد خواصّ فایبر به صورت ایزوتروپ عرضی در نظر گرفته شده است. در نتیجه از آنجا که ماتریس مدل ایزوتروپ است می‌توان کامپوزیت را ایزوتروپ عرضی فرض نمود. از علم موادّ مرکب می‌دانیم که مواد ایزوتروپیک عرضی همان مواد اورتوتروپیک بوده که رفتاری متقارن در صفحات عمود بر مختصات اصلی نشان می‌دهد. این رفتار متقارن همچنین نسبت به صفحه $X_1 = 0$ در نظر گرفته می‌شود. در واقع در این حالت بی‌نهایت صفحات متقارن الاستیک عمود بر صفحه $X_1 = 0$ فرض می‌شود. رابطه تنش و کرنش در مواد ایزوتروپ عرضی که نسبت به صفحه $X_1 = 0$ ایزوتروپیک هستند به صورت زیر است.

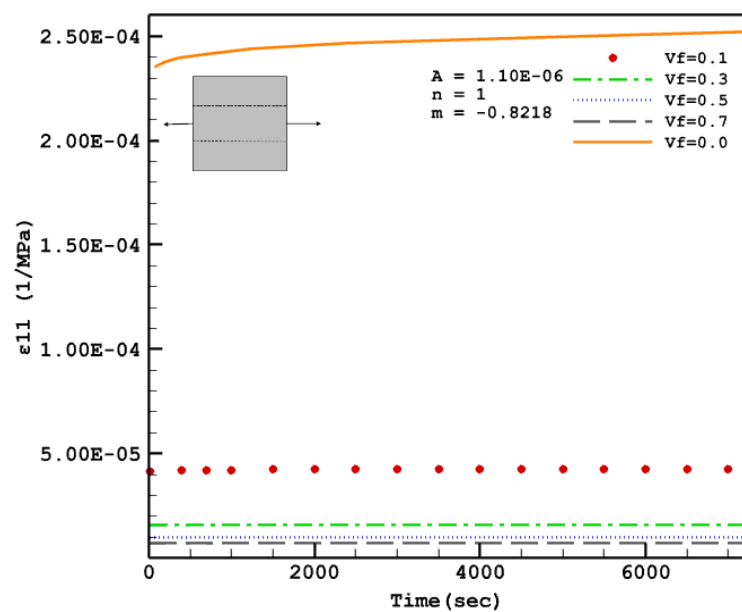
$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{12} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{12} & S_{23} & S_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{55} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} \quad (۱-۴)$$

در این حالت S_{ij} ها ضرایب ماتریس نرمی می باشند. همچنین با توجه به رابطه انرژی روابط ماتریس نرمی (کرنش) را می توان به فرم معادله (۲-۴) نوشت.

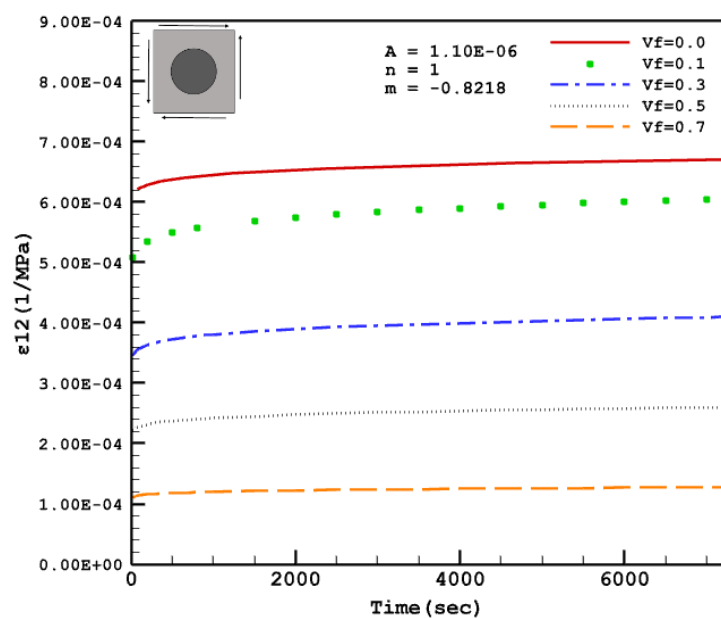
$$2U = \frac{\sigma^2}{E} \quad (۲-۴)$$

۴-۴- تاثیر نسبت حجمی بر نتایج

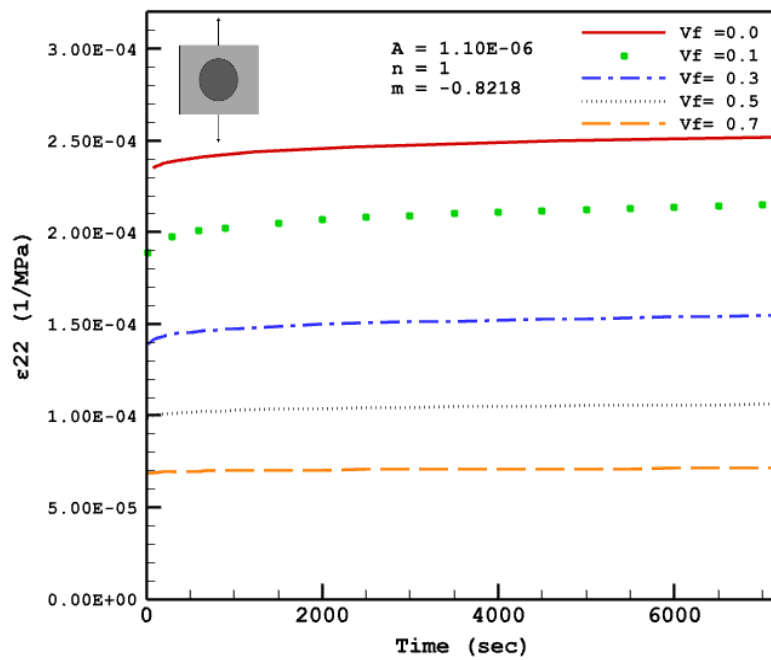
در این قسمت با افزایش نسبت حجمی از ۰/۱ تا ۰/۷ تاثیر افزایش نسبت حجمی در نتایج کامپوزیت را بررسی می شود. مانند قسمت های قبل تحلیل را برای بار ۱ مگاپاسکال و مدت زمان ۷۲۰۰ ثانیه انجام می دهیم. در شکل (۲-۴) و شکل (۳-۴) و شکل (۴-۴) و شکل (۵-۴) پنج ضریب مستقل کرنش که وابسته به زمان بوده و به صورت ضرایب موثر فرض شده اند با افزایش نسبت حجمی رسم شده اند. همانطور مشخص است با افزایش مقدار نسبت حجمی ضرایب کرنش کاهش یافته اند و از آنجا که ماتریس سفتی معکوس کرنش است در نتیجه با افزایش نسبت حجمی در واقع ماتریس سفتی بزرگتر شده و خواص کامپوزیت تقویت می شود و چون در جهت ۱ خواص ماده بیشتر تابع خواص الیاف است جهت یک خیلی دچار پدیده خزش نشده است.



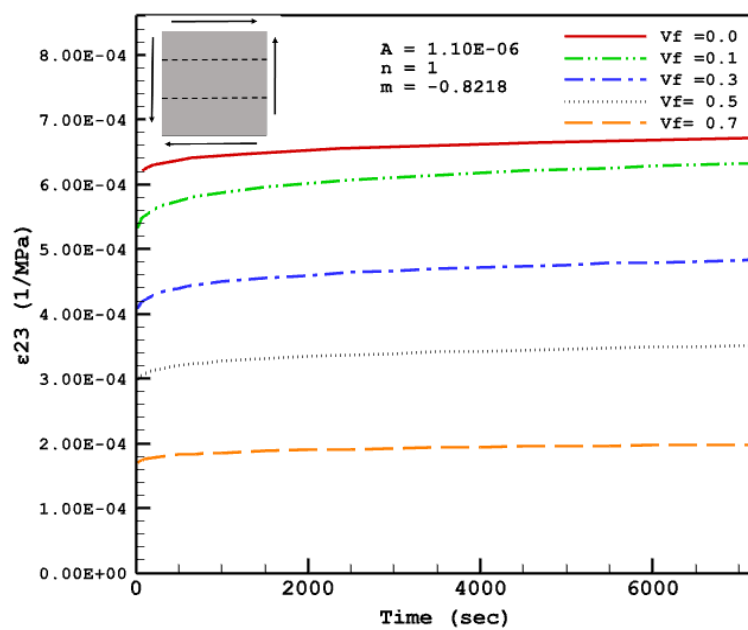
شکل (۲-۴) تاثیر نسبت حجمی بر ϵ_{11} در مدت ۲ ساعت و تحت بار ۱ مگاپاسکال



شکل (۳-۴) تاثیر نسبت حجمی بر ϵ_{12} در مدت ۲ ساعت و تحت بار ۱ مگاپاسکال

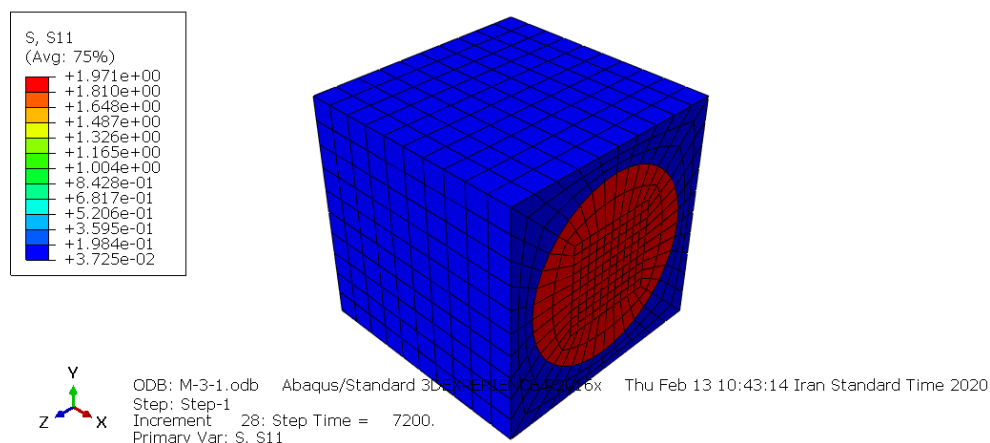
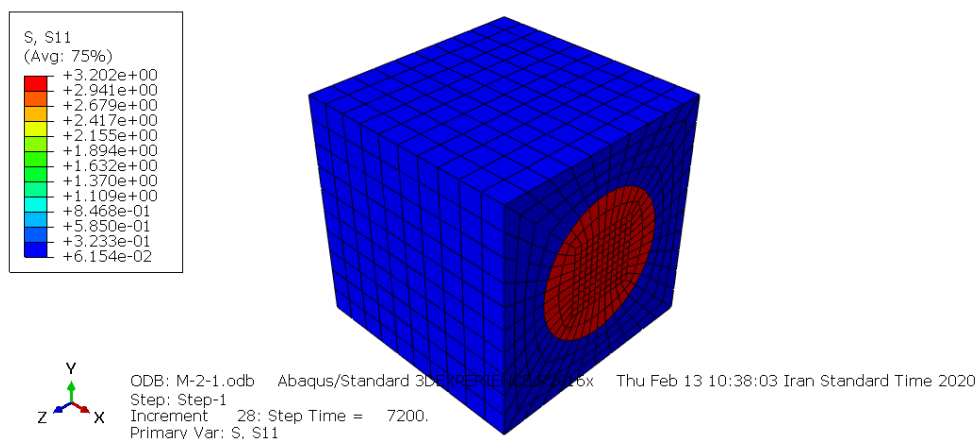
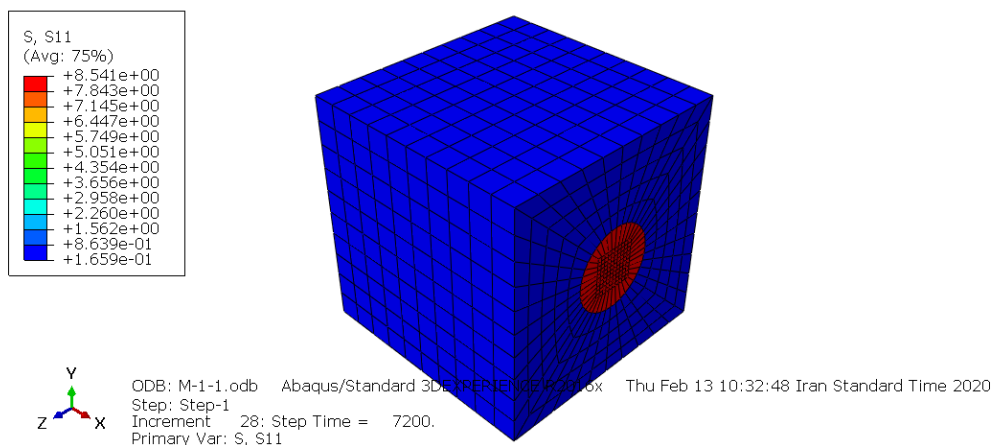


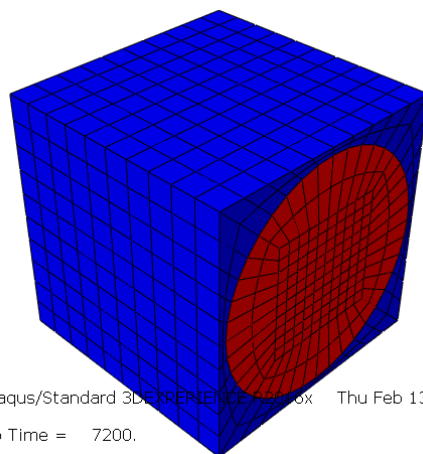
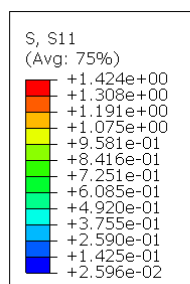
شکل (۴-۴) تاثیر نسبت حجمی بر ϵ_{22} در مدت ۲ ساعت و تحت بار ۱ مگاپاسکال



شکل (۵-۴) تاثیر نسبت حجمی بر ϵ_{23} در مدت ۲ ساعت و تحت بار ۱ مگاپاسکال

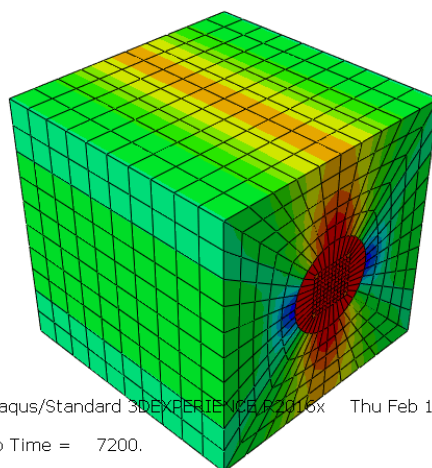
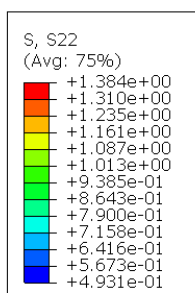
و همچنین کانتور های تنش در درصد حجمی های ۰ و ۱/۳ و ۱/۵ و ۱/۷ و در شرایط مرزی های مختلف تحت بار ۱ مگاپاسکال در مدت ۷۲۰۰ ثانیه در شکل (۴-۶) و شکل (۴-۷) و شکل (۴-۸) و شکل (۴-۹) آورده شده است. مشاهده می شود که با افزایش درصد حجمی کانتورهای تنش کاهش یافته اند.



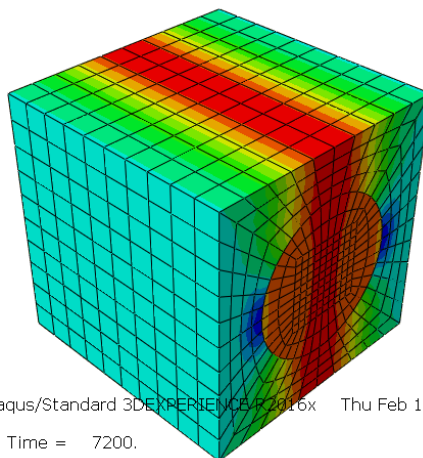
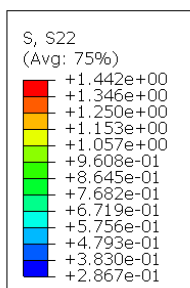


ODB: M-4-1.odb Abaqus/Standard 3DEXPERIENCE 62016x Thu Feb 13 10:48:23 Iran Standard Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Var: S, S11

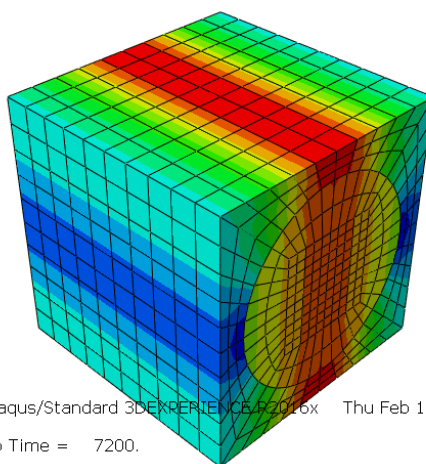
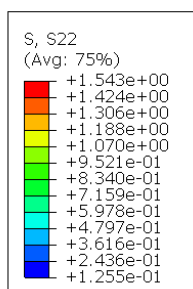
شکل (۴-۶) کانتور تنش S11 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۱ و ۰/۳ و ۰/۵ و ۰/۷



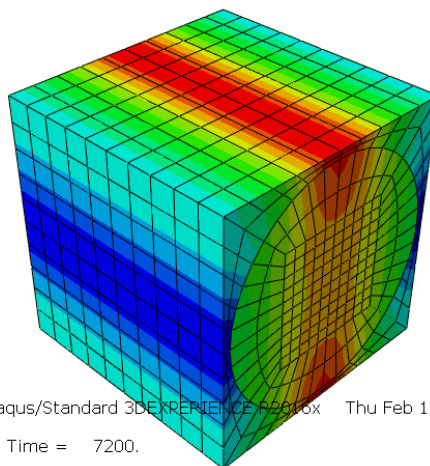
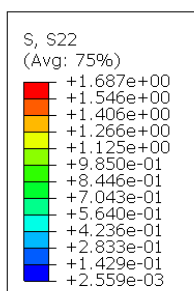
ODB: M-1-2.odb Abaqus/Standard 3DEXPERIENCE 62016x Thu Feb 13 10:34:13 Iran Standard Time 2020
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S22



ODB: M-2-2.odb Abaqus/Standard 3DEXPERIENCE 62016x Thu Feb 13 10:39:18 Iran Standard Time 2020
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S22

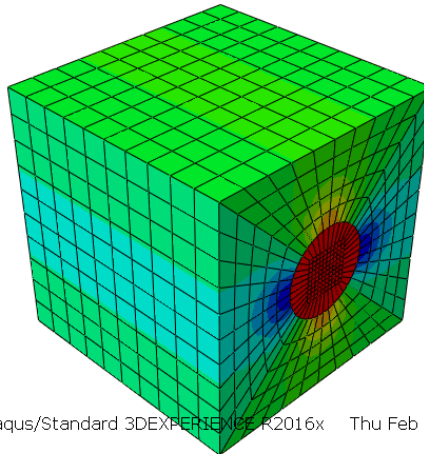
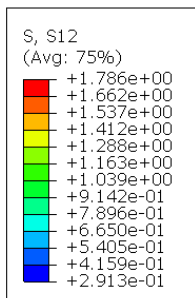


ODB: M-3-2.odb Abaqus/Standard 3DEXPERIENCE 6.20.16x Thu Feb 13 10:44:29 Iran Standard Time 2020
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S22

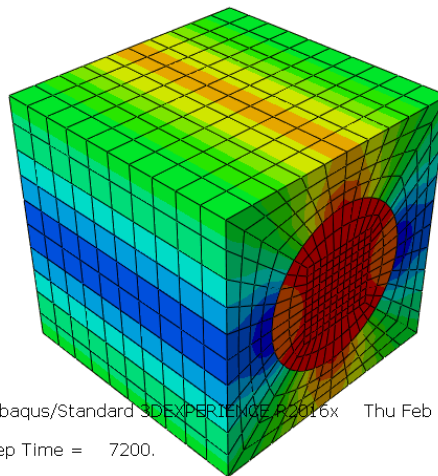
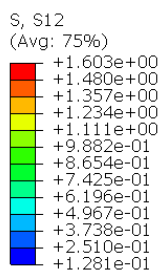


ODB: M-4-2.odb Abaqus/Standard 3DEXPERIENCE 6.20.16x Thu Feb 13 10:49:31 Iran Standard Time 2020
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S22

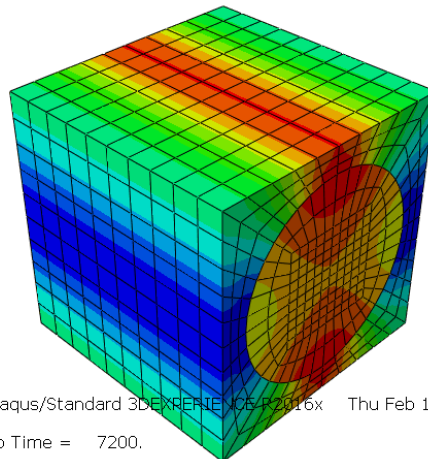
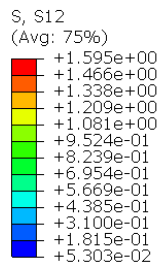
شکل (۷-۴) کانتور تنش S22 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۱ و ۰/۳ و ۰/۵ و ۰/۷



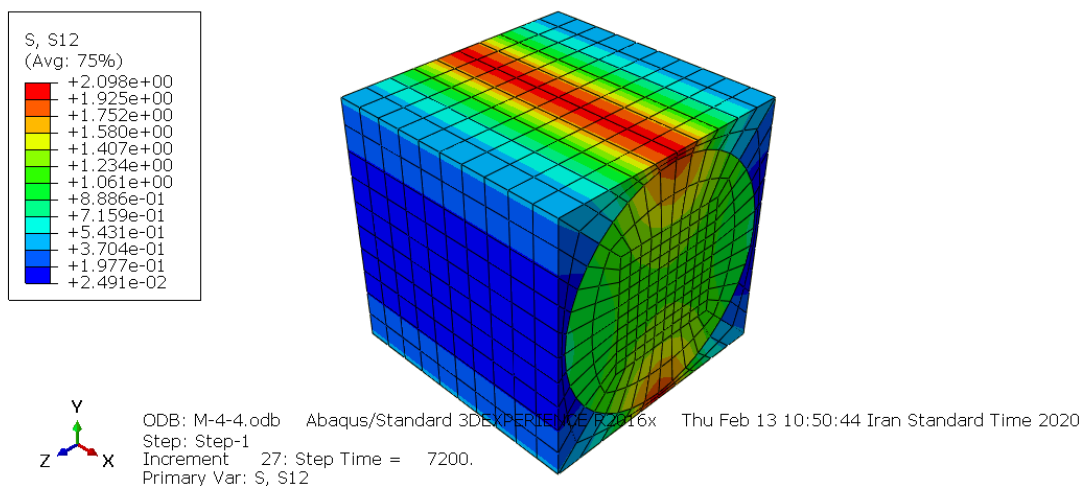
ODB: M-1-4.odb Abaqus/Standard 3DEXPERIENCE R2016x Thu Feb 13 10:35:28 Iran Standard Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Var: S, S12



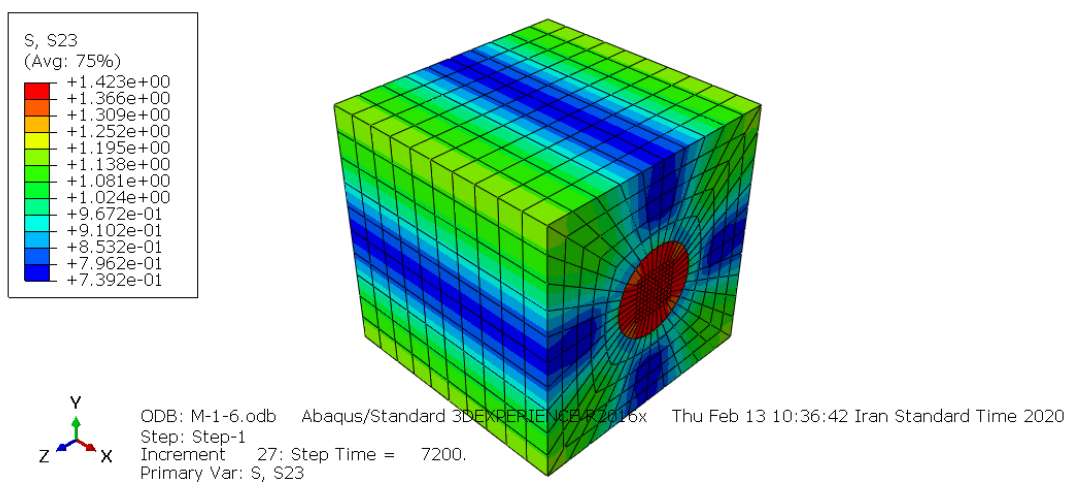
ODB: M-2-4.odb Abaqus/Standard 3DEXPERIENCE R2016x Thu Feb 13 10:40:38 Iran Standard Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Var: S, S12

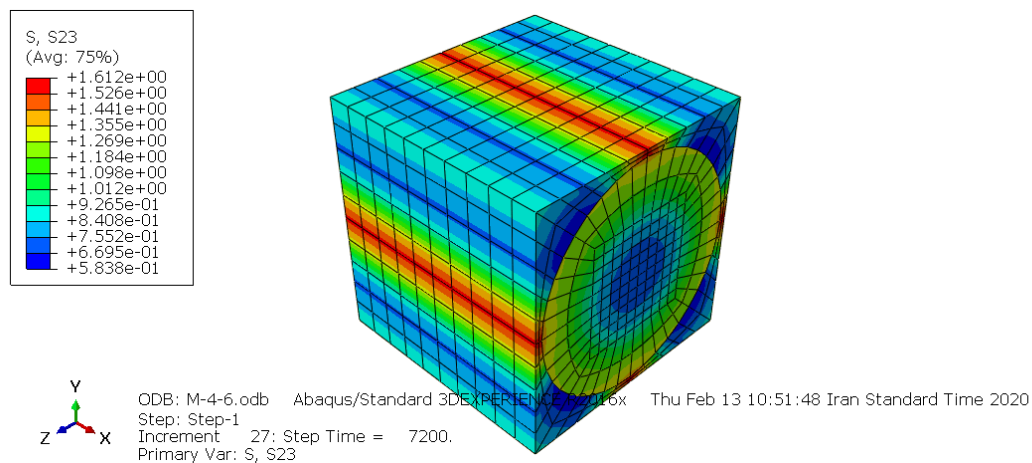
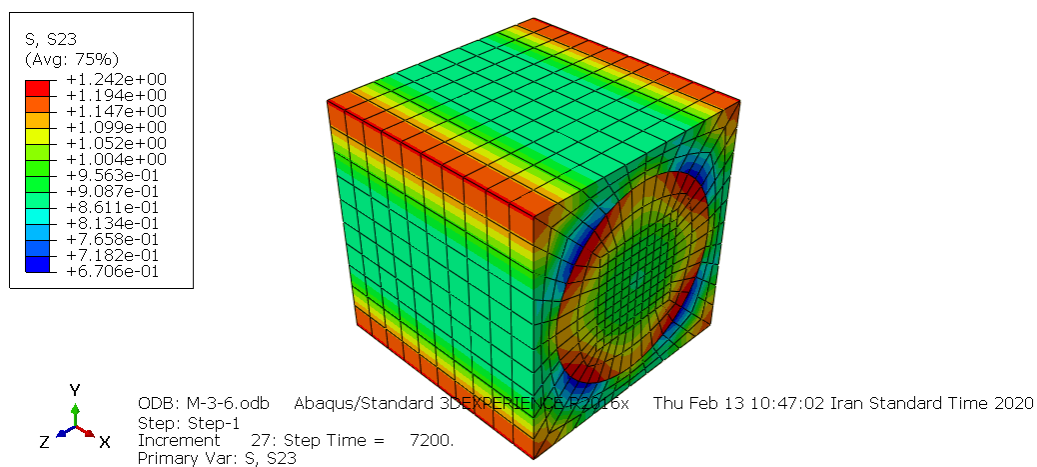
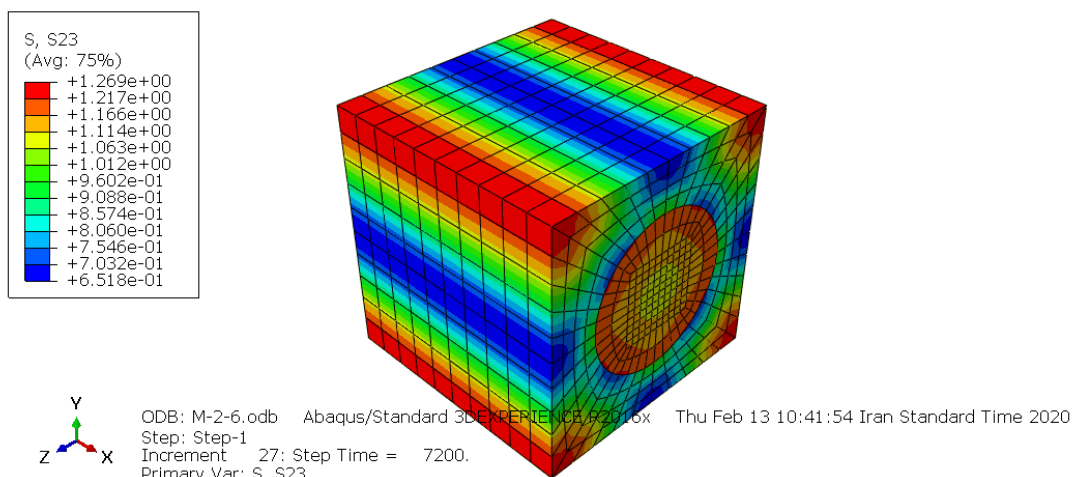


ODB: M-3-4.odb Abaqus/Standard 3DEXPERIENCE R2016x Thu Feb 13 10:45:49 Iran Standard Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Var: S, S12



شکل (۸-۴) کانتور تنش S12 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی های مختلف ۰/۷ و ۰/۵ و ۰/۳ و ۰/۱





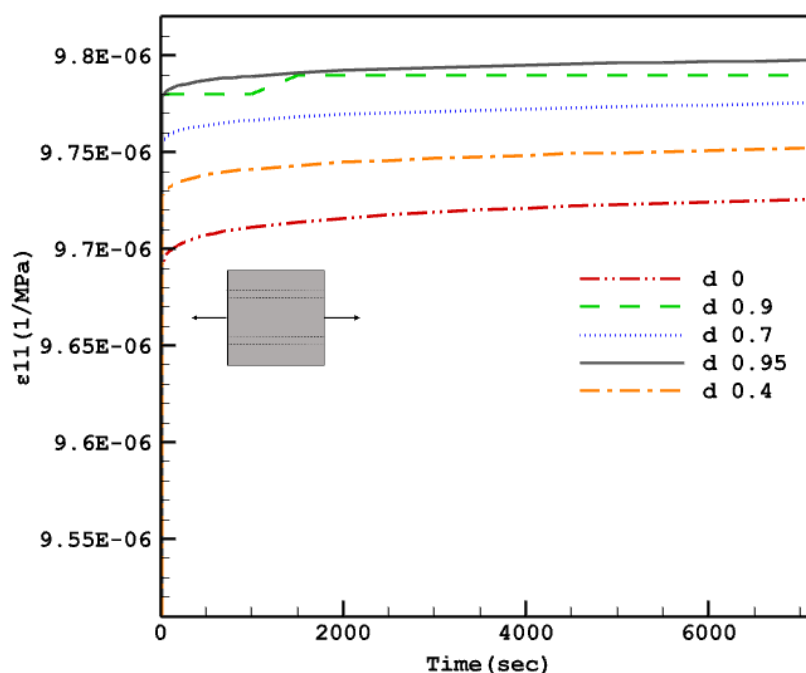
شکل (۹-۴) کانتور تنش S23 تحت بار ۱ مگاپاسکال در مدّت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۱ و ۰/۳ و ۰/۵ و ۰/۷

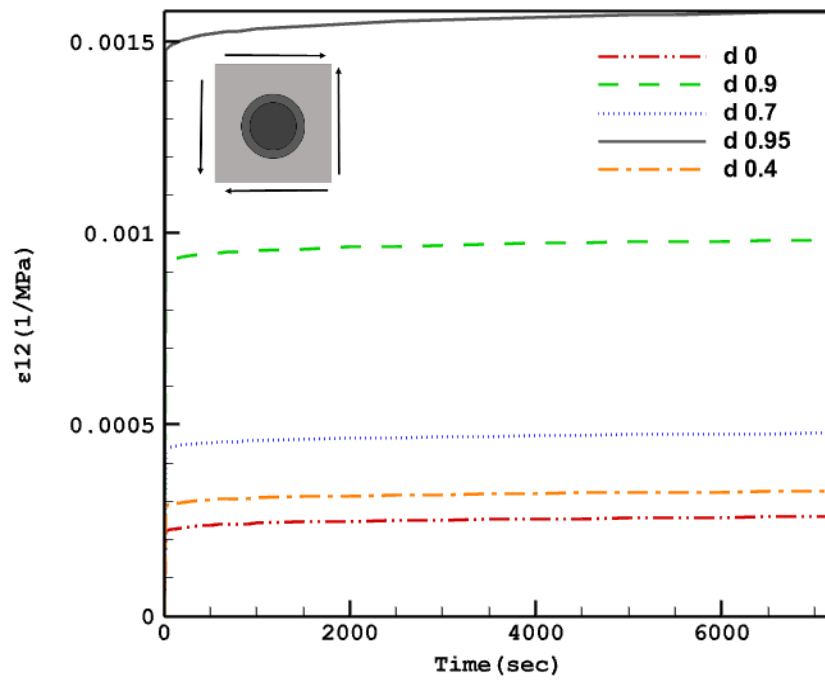
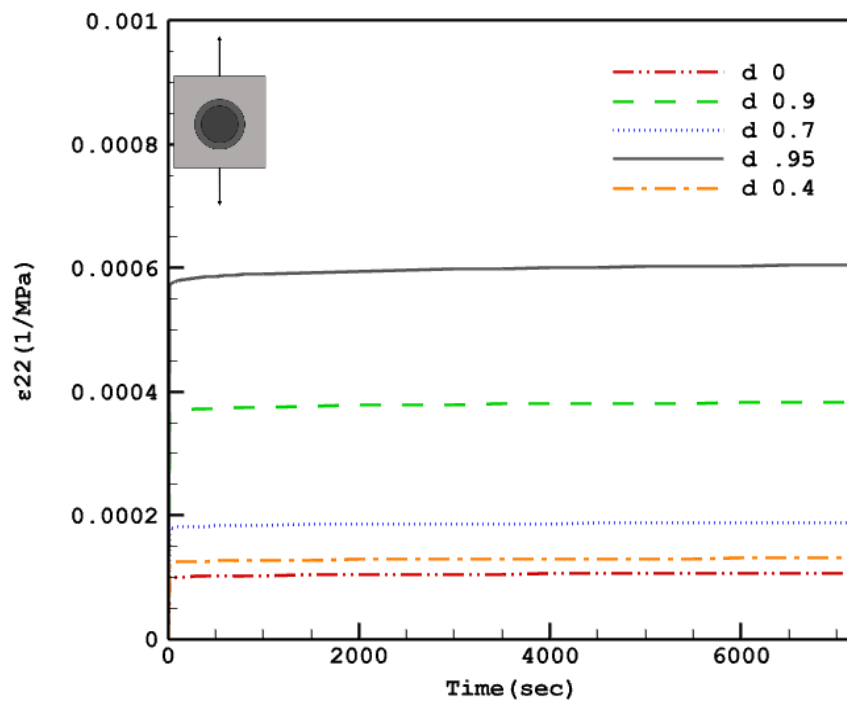
۴-۵- تاثیر نسبت حجمی بر نتایج در فاز میانی

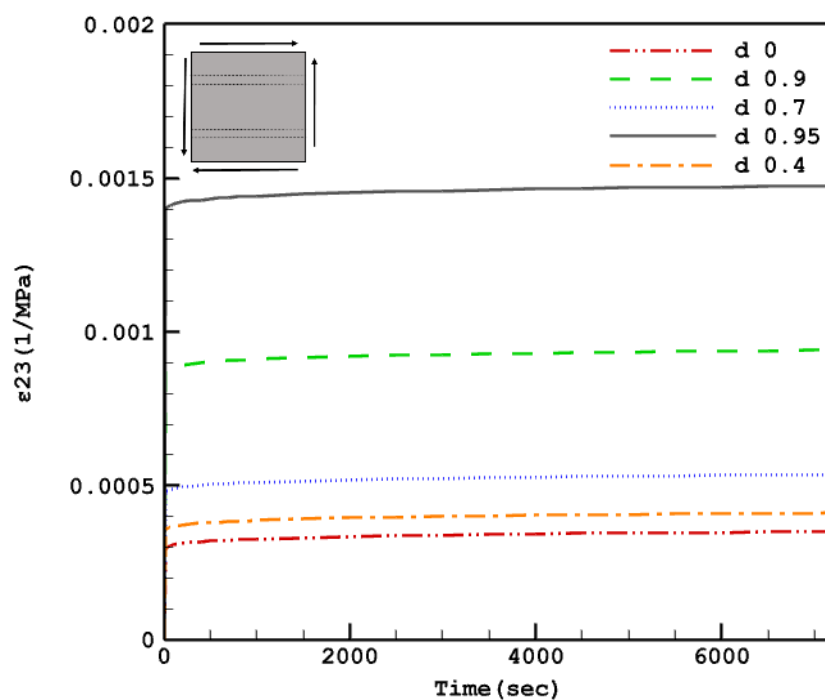
در این قسمت با نسبت حجمی ۰/۵ و در ۵ مدول الاستیسته مختلف E_m ۰/۱ و E_m ۰/۳ و E_m ۰/۵ و E_m ۰/۶ و E_m ۱ و در ضخامت فازمیانی ۰/۰۸ بررسی می‌شود. مانند قسمت‌های قبل تحلیل را برای ۱ مگاپاسکال و مدت زمان ۷۲۰۰ ثانیه انجام می‌دهیم. در شکل (۴-۱۰) چهار ضریب مستقل کرنش که وابسته به زمان بوده و به صورت ضرایب موثر فرض شده‌اند با تغییر نسبت مدول الاستیسته رسم شده‌اند. همانطور که مشخص است با کاهش مقدار نسبت مدول الاستیسته ضرایب کرنش افزایش یافته‌اند و از آنجا که ماتریس سفتی معکوس کرنش است در نتیجه با کاهش نسبت مدول الاستیسته در واقع ماتریس سفتی بزرگ‌تر شده است. همچنین یک پارامتر آسیبی به صورت معادله (۴-۳) بیان کردیم و نمودارها با توجه به این پارامتر با یکدیگر مقایسه شده‌اند. و خرابی در فازمیانی در ε_{11} تاثیرش نسبت به ε_{22} و حالت‌های برشی کمتر است.

$$d = \frac{E_m - E_i}{E_m} \quad (۴-۳)$$

E_m مدول ماتریس و E_i مدول فاز میانی می‌باشد.

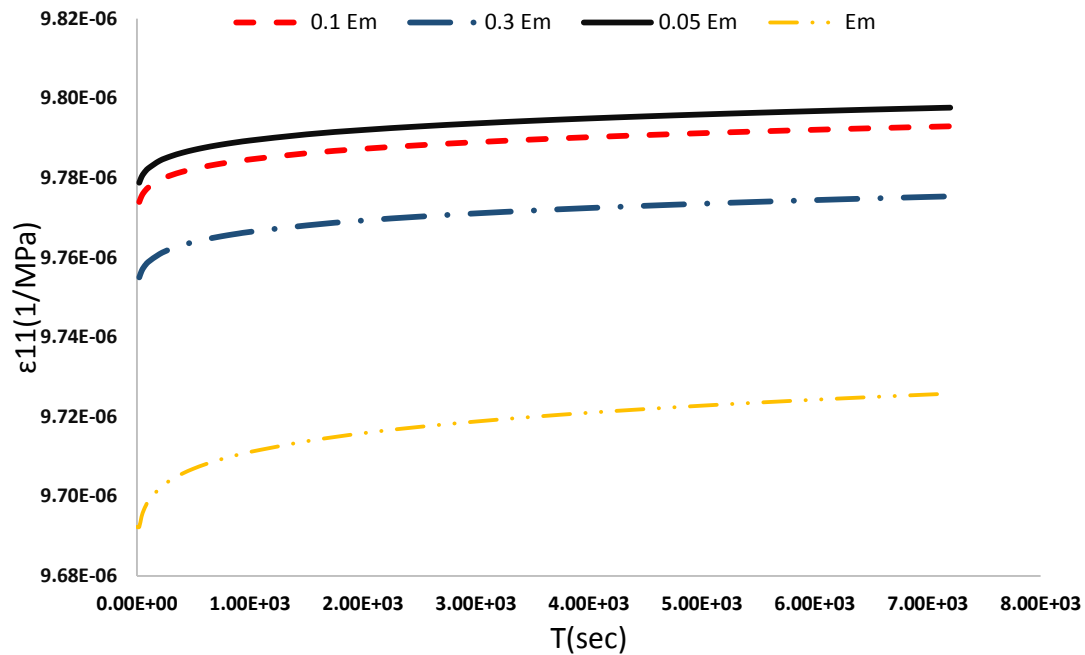






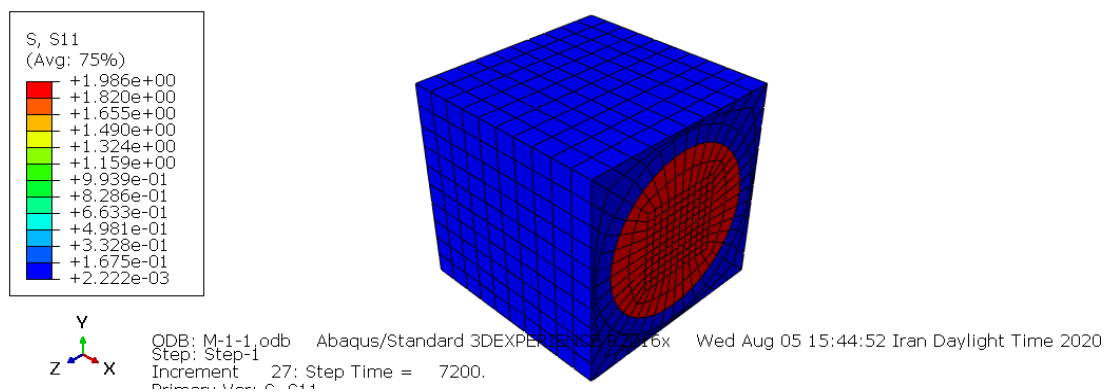
شکل (۴-۱۰) نمودار کرنش ϵ_{11} و ϵ_{22} و ϵ_{12} و ϵ_{23} در نسبت حجمی ۰/۵ و در ۵ مدول الاستیسیته مختلف $E_m 0/1$ و $E_m 0/3$ و $E_m 0/6$ و $E_m 0/5$ در ضخام فاز میانی ۰/۰۸

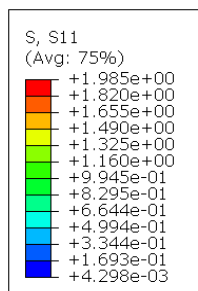
با توجه به نمودار شکل (۴-۱۱) مشاهده می‌شود که با تغییر نسبت مدول الاستیسیته مقدار کرنش نیز تغییر می‌کند. همانطور که که مشاهده می‌شود با افزایش نسبت مدول الاستیسیته مقدار کرنش نیز به طبع کمتر می‌شود به دلیل اینکه مدول الاستیسیته رابطه عکسی با کرنش دارد.



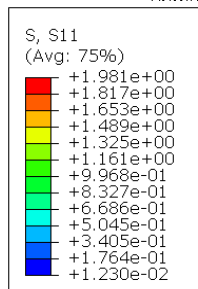
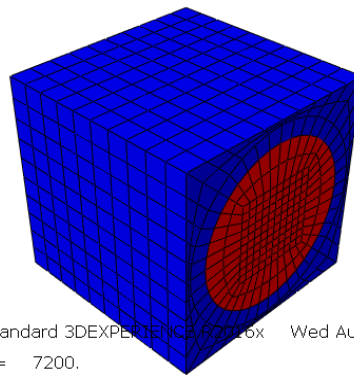
شکل (۴-۱۱) مقایسه حالت تک‌فایبر با تغییر درصد مدول الاستیسیته در حالت دارای فازمیانی

و همچنین کانتورهای تنش در درصد حجمی ۰٫۵ و در ۳ مدول الاستیسیته مختلف $Em/۱$ و $Em/۳$ و $Em/۰٫۵$ و $Em/۰٫۶$ و در خود Em در ضخامت فازمیانی $۰٫۰۸/۰$ در شرایط مرزی‌های مختلف تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه در شکل (۴-۱۲) و شکل (۴-۱۳) و شکل (۴-۱۴) و شکل (۴-۱۵) آورده شده‌است. همانطور که مشاهده می‌شود در حالت‌های $S11$ و $S22$ و $S23$ با افزایش مدول الاستیسیته تنش افزایش یافته و در حالت $S12$ با افزایش مدول الاستیسیته تنش کاهش یافته‌است.

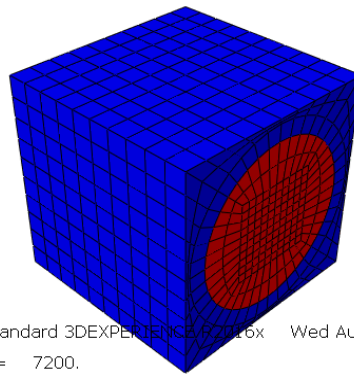




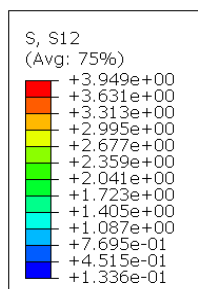
ODB: M-1-1.odb Abaqus/Standard 3DEXPERIENCE 6/20/16x Wed Aug 05 15:36:46 Iran Daylight Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Var: S, S11



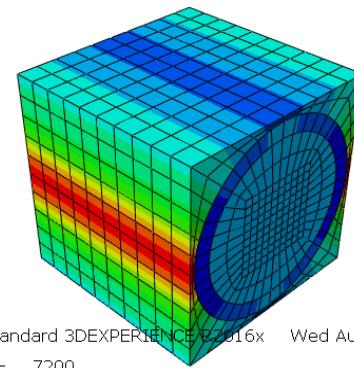
ODB: M-1-1.odb Abaqus/Standard 3DEXPERIENCE 6/20/16x Wed Aug 05 15:52:55 Iran Daylight Time 2020
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S11

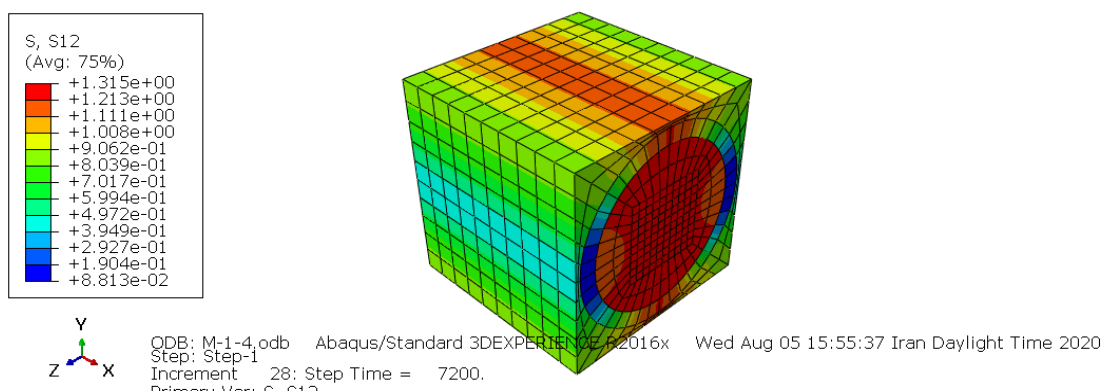
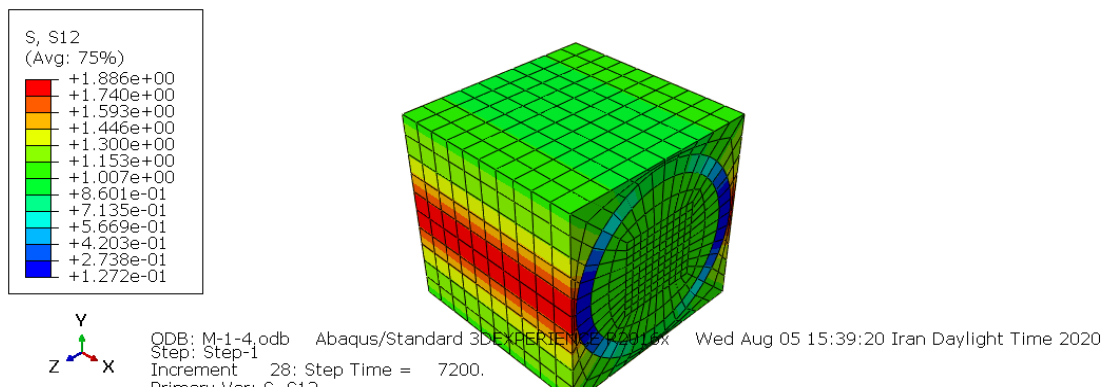


شکل (۴-۱۲) کانتور تنش S11 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه در حالت فاز میانی و به ترتیب در ۰/۵ و ۰/۱ و ۰/۳ و ۰/۶ و ۱ مدول الاستیسیته

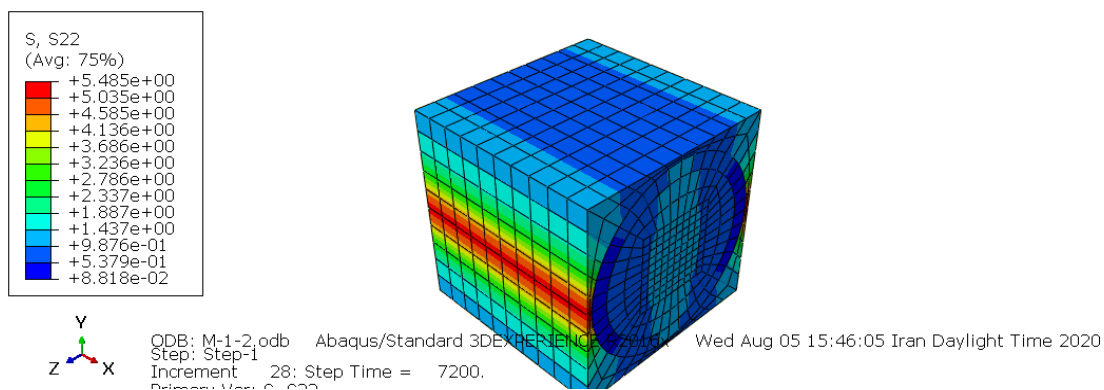


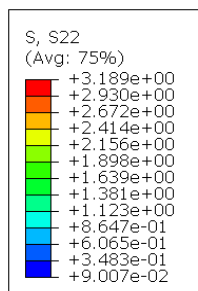
ODB: M-1-4.odb Abaqus/Standard 3DEXPERIENCE 6/20/16x Wed Aug 05 15:47:25 Iran Daylight Time 2020
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S12



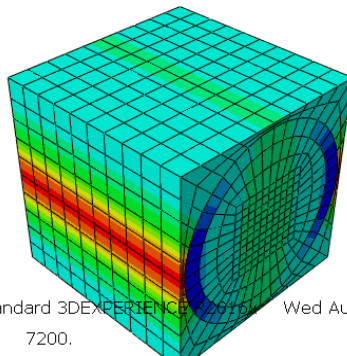


شکل (۴-۱۳) کانتور تنش S12 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه در حالت فاز میانی و به ترتیب در ۰/۵ و ۰/۱ و ۰/۳ و ۰/۶ و ۱ مدول الاستیسیته

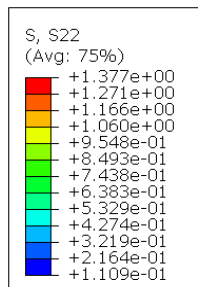




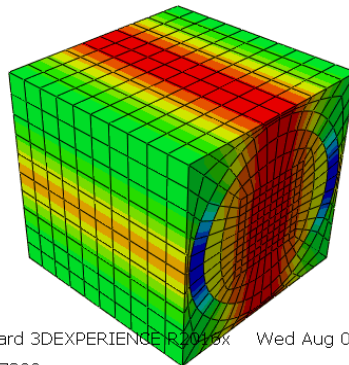
ODB: M-1-2.odb Abaqus/Standard 3DEXPERIENCE R2016x
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S22



Wed Aug 05 15:38:00 Iran Daylight Time 2020

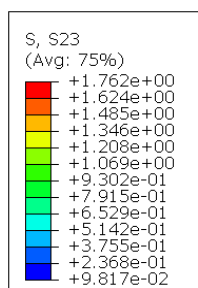


ODB: M-1-2.odb Abaqus/Standard 3DEXPERIENCE R2016x
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S22

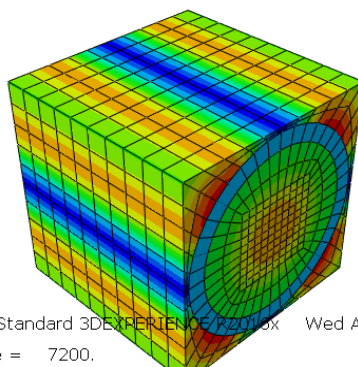


Wed Aug 05 15:54:11 Iran Daylight Time 2020

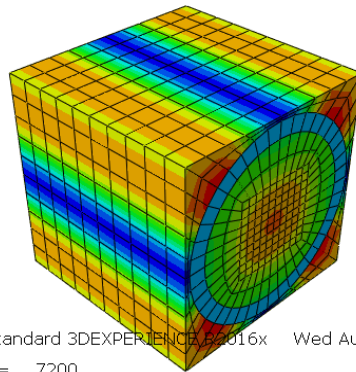
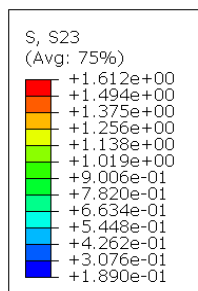
شکل (۴-۱۴) کانتور تنش S22 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه در حالت فاز میانی و به ترتیب در ۰/۵ و ۰/۱ و ۰/۳ و ۰/۶ و ۱۰ مدول الاستیسیته



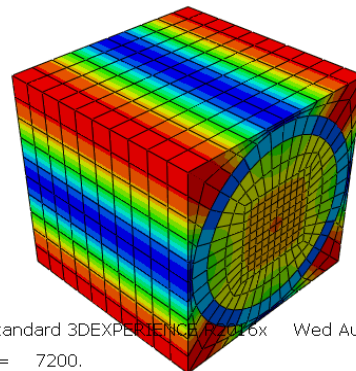
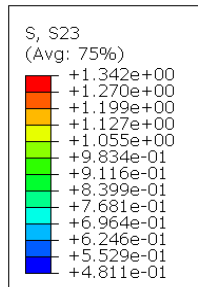
ODB: M-1-6.odb Abaqus/Standard 3DEXPERIENCE R2016x
Step: Step-1
Increment 28: Step Time = 7200.
Primary Var: S, S23



Wed Aug 05 15:48:49 Iran Daylight Time 2020



ODB: M-1-6.odb Abaqus/Standard 3DEXPERIENCE 2016x Wed Aug 05 15:40:32 Iran Daylight Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Variable: S23

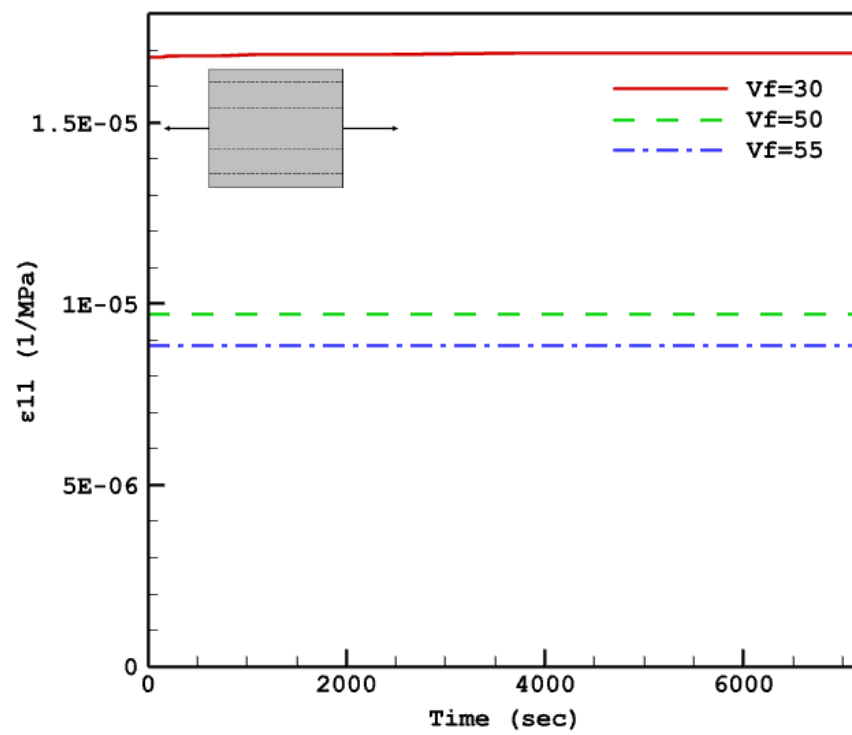


ODB: M-1-6.odb Abaqus/Standard 3DEXPERIENCE 2016x Wed Aug 05 15:57:00 Iran Daylight Time 2020
Step: Step-1
Increment 27: Step Time = 7200.
Primary Variable: S23

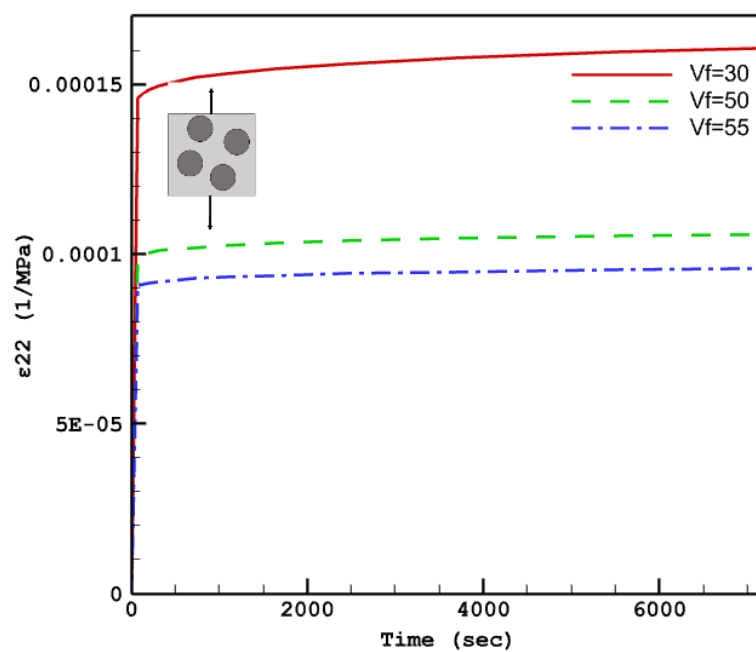
شکل (۴-۱۵) کانتور تنش S23 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه در حالت فاز میانی و به ترتیب در ۰/۵ و ۰/۱ و ۰/۳ و ۰/۶ و ۱۰/۱ مدول الاستیسیته

۴-۶- نتایج توزیع تصادفی الیاف

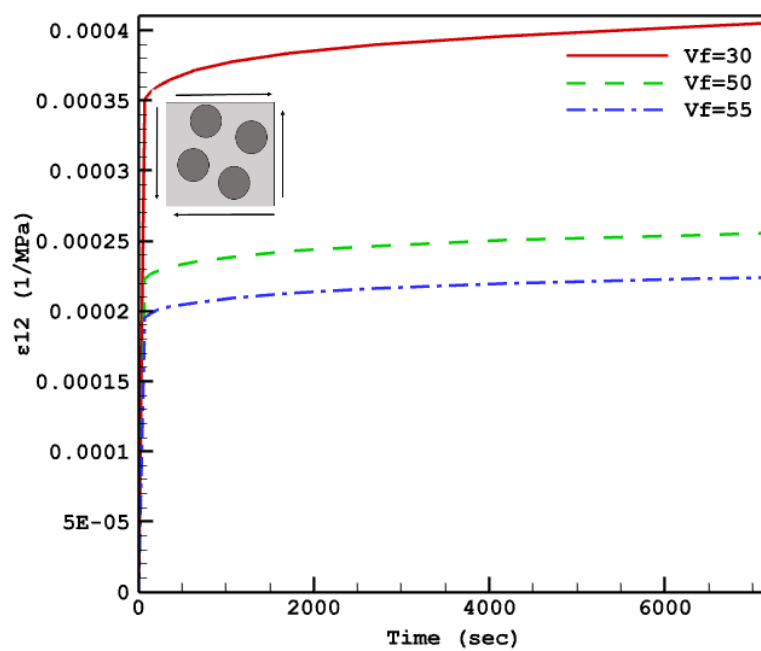
در این قسمت با نسبت حجمی ۵۵/۰ و ۵/۰ و ۳/۰ و در اندازه شبکه بندی ۱/۰ و ۴ شرایط مرزی انجام شده است. مانند قسمت های قبل تحلیل را برای بار ۱ مگاپاسکال و مدت زمان ۷۲۰۰ ثانیه انجام می دهیم. در شکل (۴-۱۶) (شکل (۴-۱۷) و شکل (۴-۱۸) و شکل (۴-۱۹) چهار ضریب مستقل کرنش که وابسته به زمان بوده و به صورت ضرایب موثر فرض شده اند که با تغییر نسبت حجمی رسم شده اند آورده شده است. همانطور مشخص است با کاهش مقدار نسبت حجمی ضرایب کرنش افزایش یافته اند و از آنجا که ماتریس سفتی معکوس کرنش است در نتیجه با کاهش نسبت حجمی در واقع ماتریس سفتی بزرگ تر شده و خواص کامپوزیت تقویت شده است.



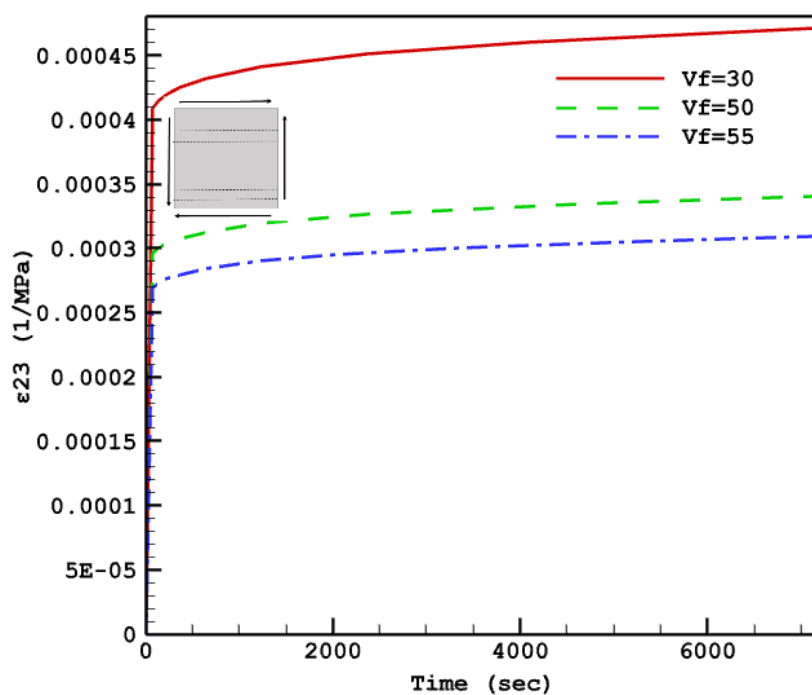
شکل (۴-۱۶) نمودار کرنش ϵ_{11} تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی های مختلف ۰/۳ و ۰/۵ و ۰/۵۵



شکل (۴-۱۷) نمودار کرنش ϵ_{22} تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی های مختلف ۰/۳ و ۰/۵ و ۰/۵۵

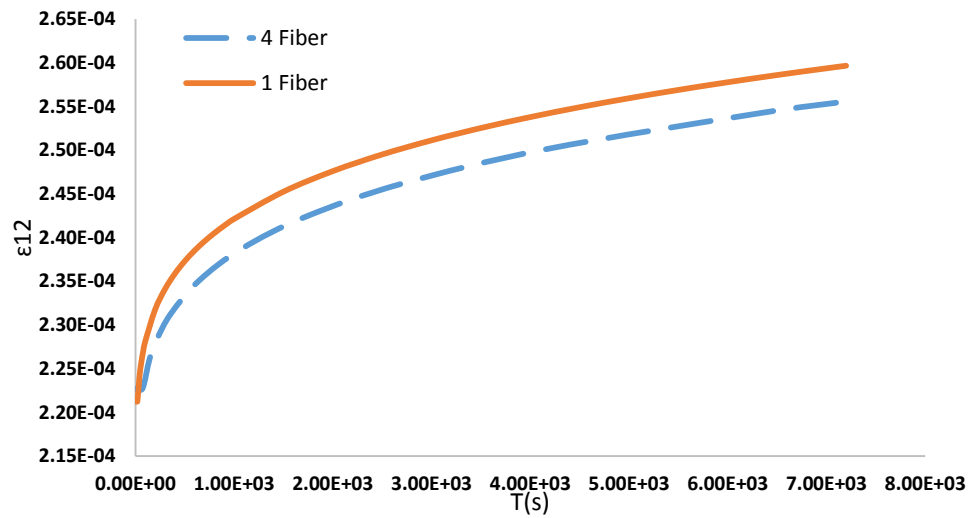


شکل (۴-۱۸) نمودار کرنش ε_{12} تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی های مختلف ۰/۳ و ۰/۵ و ۰/۵۵



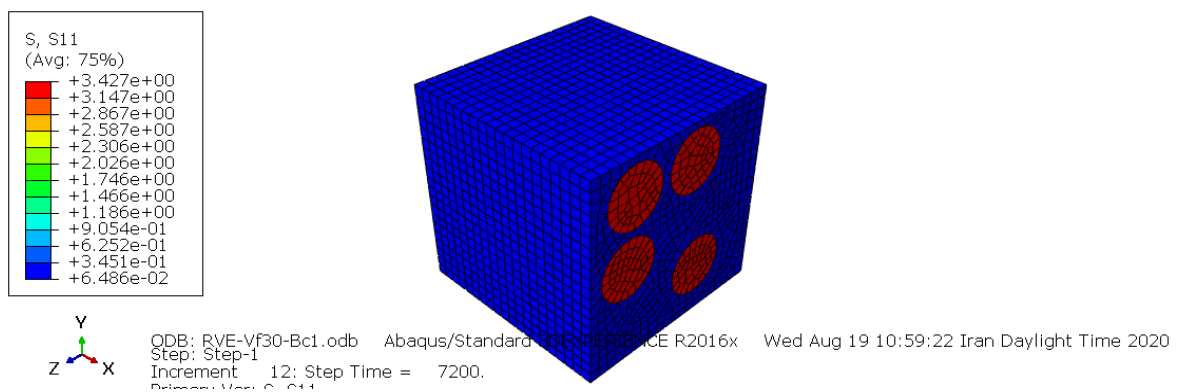
شکل (۴-۱۹) نمودار کرنش ε_{23} تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی های مختلف ۰/۳ و ۰/۵ و ۰/۵۵

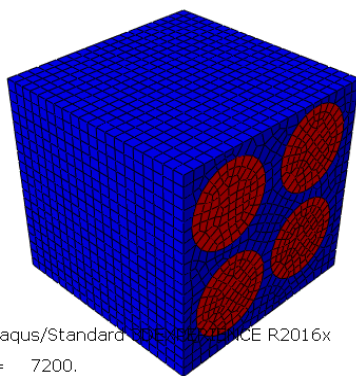
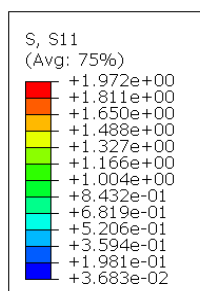
با استفاده از الگوریتم ارائه شده برای مدل تولید تصادفی الیاف و فراخوانی زیربرنامه نوشته شده، به حل آن پرداخته شده است. نوع مدل و شرایط مرزی آن به صورتی در نظر گرفته شده که متناسب با شرایط مدل تکفایبر بوده است. پس از انجام تحلیل و با مقایسه کردن مدل ران شده با مدل تکفایبر که در شکل (۴-۲۰) قابل مشاهده است می توان تطابق خوبی بین مدل تکفایبر و مدل ۴ فایبر مشاهده کرد.



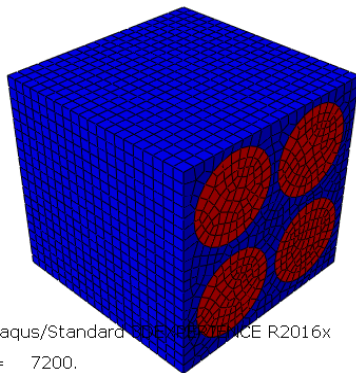
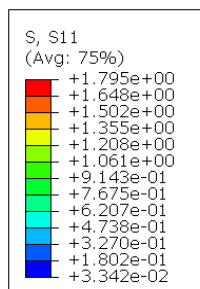
شکل (۴-۲۰) مقایسه کردن مدل ۴ فایبر با مدل تک فایبر

و همچنین کانتورهای تنش در درصد حجمی‌های ۰/۵ و ۰/۳ و ۰/۱ در شرایط مرزی‌های مختلف تحت بار ۱ و مگاپاسکال در مدت ۷۲۰۰ ثانیه در شکل (۴-۲۱) و شکل (۴-۲۲) و شکل (۴-۲۳) و شکل (۴-۲۴) آورده شده‌است. همانطور که مشاهده می‌شود در حالت‌های S_{11} و S_{22} و S_{12} با افزایش درصد حجمی تنش کاهش یافته اما در S_{23} با افزایش درصد حجمی تنش افزایش یافته است.



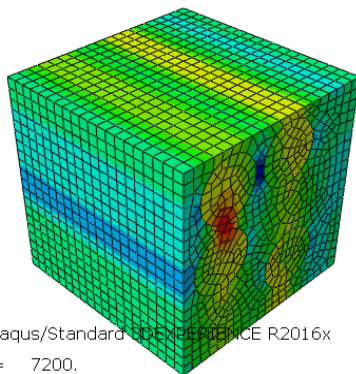
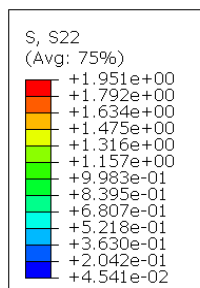


ODB: RVE-Vf50-Bc1.odb Abaqus/Standard 6.14-1 EXPLICIT R2016x Wed Aug 19 11:13:49 Iran Daylight Time 2020
Step: Step-1
Increment 12: Step Time = 7200.
Primary Var: S, S11

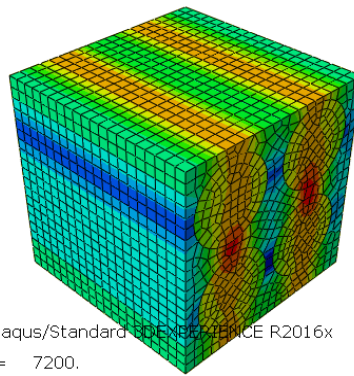
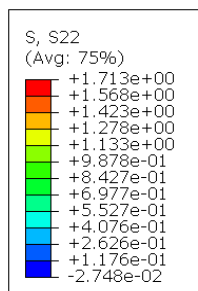


ODB: RVE-Vf55-Bc1.odb Abaqus/Standard 6.14-1 EXPLICIT R2016x Wed Aug 19 11:26:40 Iran Daylight Time 2020
Step: Step-1
Increment 12: Step Time = 7200.
Primary Var: S, S11

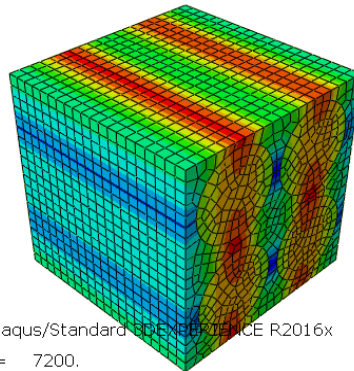
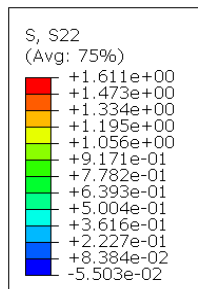
شکل (۴-۲۱) کانتور تنش S11 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۵۵ و ۰/۵ و ۰/۳



ODB: RVE-Vf30-Bc2.odb Abaqus/Standard 6.14-1 EXPLICIT R2016x Wed Aug 19 11:03:19 Iran Daylight Time 2020
Step: Step-1
Increment 12: Step Time = 7200.
Primary Var: S, S22

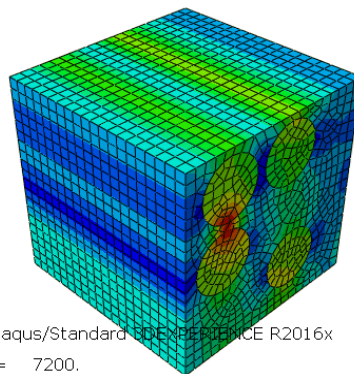
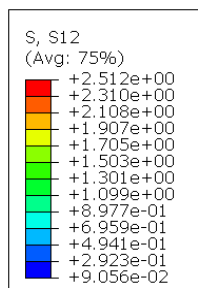


ODB: RVE-Vf50-Bc2.odb Abaqus/Standard ODB EXPTENCE R2016x Wed Aug 19 11:17:11 Iran Daylight Time 2020
Step: Step-1
Increment 12: Step Time = 7200.
Primary Var: S, S22

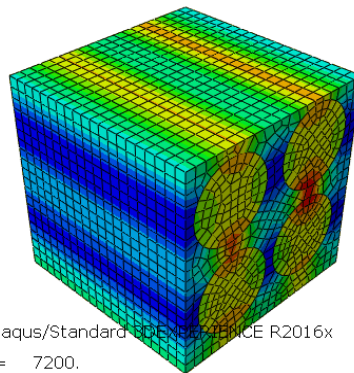
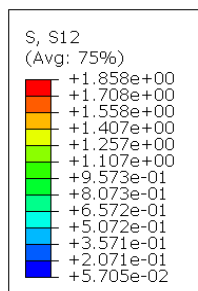


ODB: RVE-Vf55-Bc2.odb Abaqus/Standard ODB EXPTENCE R2016x Wed Aug 19 11:29:49 Iran Daylight Time 2020
Step: Step-1
Increment 12: Step Time = 7200.
Primary Var: S, S22

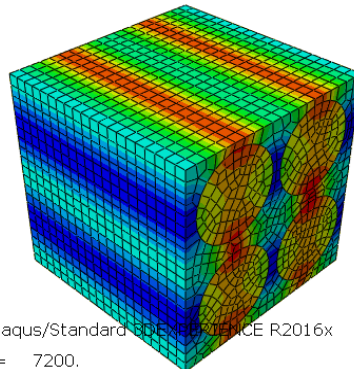
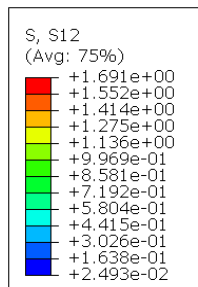
شکل (۴-۲۲) کانتور تنش S22 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۳ و ۰/۵ و ۰/۵۵



ODB: RVE-Vf30-Bc4.odb Abaqus/Standard ODB EXPTENCE R2016x Wed Aug 19 11:06:52 Iran Daylight Time 2020
Step: Step-1
Increment 11: Step Time = 7200.
Primary Var: S, S12

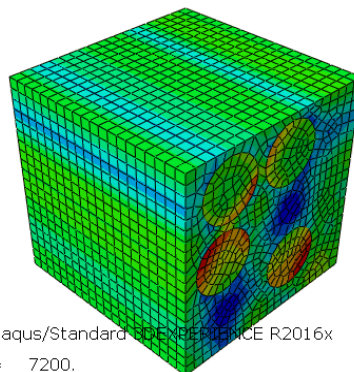
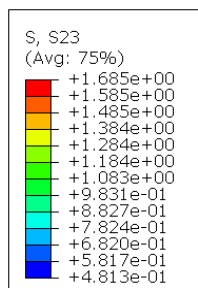


ODB: RVE-Vf50-Bc4.odb Abaqus/Standard ODB EXPLORER R2016x Wed Aug 19 11:20:36 Iran Daylight Time 2020
Step: Step-1
Increment 11: Step Time = 7200.
Primary Var: S, S12

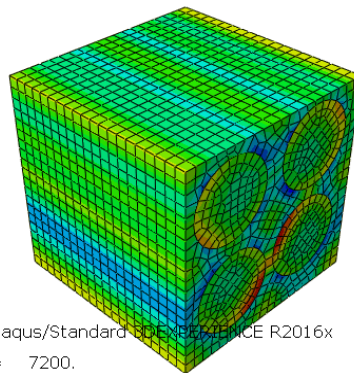
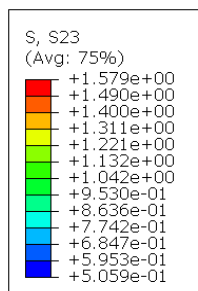


ODB: RVE-Vf55-Bc4.odb Abaqus/Standard ODB EXPLORER R2016x Wed Aug 19 11:33:03 Iran Daylight Time 2020
Step: Step-1
Increment 11: Step Time = 7200.
Primary Var: S, S12

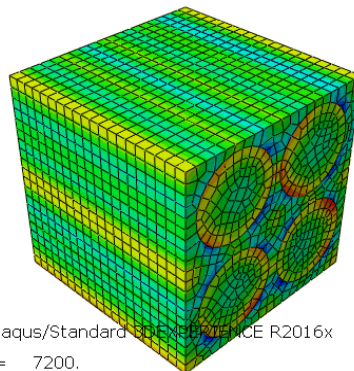
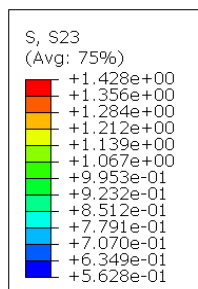
شکل (۴-۲۳) کانتور تنش S12 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۳ و ۰/۵ و ۰/۵۵



ODB: RVE-Vf30-Bc6.odb Abaqus/Standard ODB EXPLORER R2016x Wed Aug 19 11:10:18 Iran Daylight Time 2020
Step: Step-1
Increment 9: Step Time = 7200.
Primary Var: S, S23



ODB: RVE-Vf50-Bc6.odb Abaqus/Standard ODB EXPLORER R2016x Wed Aug 19 11:23:33 Iran Daylight Time 2020
Step: Step-1
Increment 9; Step Time = 7200.
Primary Var: S, S23



ODB: RVE-Vf55-Bc6.odb Abaqus/Standard ODB EXPLORER R2016x Wed Aug 19 11:36:03 Iran Daylight Time 2020
Step: Step-1
Increment 10; Step Time = 7200.
Primary Var: S, S23

شکل (۴-۲۴) کانتور تنش S23 تحت بار ۱ مگاپاسکال در مدت زمان ۷۲۰۰ ثانیه به ترتیب در نسبت حجمی‌های مختلف ۰/۵۵ و ۰/۵ و ۰/۳

۴-۷- جمع بندی

در این فصل به صحنه گذاری مدل پرداخته شد و در ادامه نتایج برای سه حالت تک فایبر و تک فایبر دارای فاز میانی و حالت توزیع تصادفی الیاف بررسی شد.

فصل ۵:

جمع‌بندی و پیشنهادها

۵-۱- مقدمه

استفاده از قابلیت‌های مواد کامپوزیتی در طراحی سازه‌های هدفمند در صنایع متعدد در حال گسترش است و در مواد کامپوزیتی از زمینه‌های پلیمری به عنوان فاز نگهدارنده استفاده می‌کنند به دلیل رفتار ویسکوالاستیک مواد زمینه دچار پدیده‌های آزادسازی تنش و خزش حتی در دماهای پایین می‌شوند به همین دلیل لحاظ کردن پدیده‌های فوق در طراحی سازه‌های کامپوزیتی از اهمیت ویژه‌ای برخوردار است یکی از چالش برانگیزترین مسائل در بحث خزش کامپوزیت‌ها ارائه راهکاری برای پیش‌بینی رفتار خزشی آن‌ها براساس مدلسازی تئوری به منظور پرهیز از آزمون‌های درازمدت و وقت‌گیر می‌باشد. راهکارهای متفاوتی برای پیش‌بینی رفتار خزش مواد وجود دارد که بسته به نوع ماده مورد مطالعه متفاوت است. مواد پلیمری به دلیل انعطاف پذیری و وزن کم به طور گسترده در بسیاری از سازه‌های کاربردی مورد استفاده قرار می‌گیرد. رفتار این مواد معمولاً به زمان حالت بارگذاری، دما و رطوبت وابسته است. کامپوزیت‌های زمینه پلیمری نیز به سبب وجود ماتریس ویسکوالاستیک در آن‌ها، رفتاری مشابه از خود نشان می‌دهد.

در این پژوهش در فاز میکرومکانیک به بسط مدل توان پرداختیم. که با استفاده از یک کد پایتون به تولید سه مدل تک‌فایبر که فایبر به صورت دایره‌ای درون ماتریس قرار دارد و در درصد حجمی‌های ۰ و ۱/۰ و ۳/۰ و ۵/۰ و ۷/۰ صورت گرفته است و مدل تک‌فایبر دارای فازمیانی که در این حالت هم فایبر به صورت دایره‌ای درون ماتریس در درصد حجمی ۵/۰ و یک فازمیانی به ضخامت ۸/۰۰ و در سه مدول مختلف که بین فایبر و ماتریس قرار دارد قرار گرفته است. و مدل تولید تصادفی ۴ الیاف در ماتریس در درصد حجمی‌های ۵۵/۰ و ۵/۰ و ۳/۰ پرداخته شد. و همچنین برای هر سه مدل از شرایط مرزی متناوب در مدت زمان ۷۲۰۰ ثانیه و تحت بار ۱ مگاپاسکال استفاده شده است.

برای صحت سنجی از نمودار [43] استفاده شد که مشاهده گردید که تطابق خوبی بین نتایج تجربی و نتایج به دست آمده حالت تک‌فایبر وجود داشت. و نتایج حالت تک‌فایبر و حالت توزیع تصادفی نشان داد که

با افزایش نسبت حجمی الیاف کرنش کمتر می‌شود و ماتریس سفتی افزایش میابد. برای حالت فازمیانی مشاهده شد که با کاهش مقدار نسبت مدول الاستیسیته ضرایب کرنش افزایش یافته‌اند و از آنجا که ماتریس سفتی معکوس کرنش است در نتیجه با کاهش نسبت مدول الاستیسیته در واقع ماتریس سفتی بزرگ‌تر شده‌است.

۵-۲- پیشنهادها

برای ادامه‌ی این پژوهش پیشنهاد می‌شود:

- تمام مراحل صورت گرفته در این پژوهش با انجام تست‌هایی تجربی مقایسه کرد و دقت آن مورد بررسی قرار گیرد.
- مدل ارائه شده برای مسئله در ابعاد ماکرو اعمال شده و نتایج با داده‌های تجربی مقایسه گردد.
- در مدل میکرومکانیک با در نظر گرفتن ترک، ضرایب کرنش در تحلیل خزش محاسبه گردد.
- معادلات برای حالتی که اعمال بار به صورت جابجایی انجام می‌گردد بازنویسی شود.

مراجع

- [1] I. Ahmadi and N. Ataee, "modares mechanical engineering" composite materials," vol. 16, no. 8, pp. 249–260, 2016.
- [2] G. Lubin, Handbook of composites. Springer Science & Business Media, 2013.
- [3] R. F. Gibson, Principles of composite material mechanics. CRC press, 2016.
- [4] J. Stolz, P. Fideu, and A. Herrmann, "Homogenization of Fiber Composite Material Properties: an Adaptive Multiphysics Implementation."
- [5] "Rui Miranda Guedes - Creep and Fatigue in Polymer Matrix Composites (Woodhead Publishing in Materials) (2011, Woodhead Publishing) (1).pdf." .
- [6] F. Mainardi and G. Spada, "Creep, relaxation and viscosity properties for basic fractional models in rheology," European Physical Journal: Special Topics, vol. 193, no. 1, pp. 133–160, 2011, doi: 10.1140/epjst/e2011-01387-1.
- [7] D. N. Robinson, W. K. Binienda, and M. B. Ruggles, "Creep of polymer matrix composites. I: Norton/Bailey creep law for transverse isotropy," J. Eng. Mech., vol. 129, no. 3, pp. 310–317, 2003.
- [8] D. L. May, A. P. Gordon, and D. S. Segletes, "The application of the norton-bailey law for creep prediction through power law regression," Proceedings of the ASME Turbo Expo, vol. 7 A, 2013, doi: 10.1115/GT2013-96008.
- [9] P. Press, "Creep and Failure of a Full-Size Plastic Pultruded Frame +," Composites, vol. 2, no. 3, 1992.
- [10] R. A. Schapery, "an Engineering Theory of Nonlinear," International Journal of Solids and Structures, vol. 2, no. 3, pp. 407–425, 1966, [Online]. Available: [http://dx.doi.org/10.1016/0020-7683\(66\)90030-8](http://dx.doi.org/10.1016/0020-7683(66)90030-8).
- [11] A. Naik, N. Abolfathi, G. Karami, and M. Ziejewski, "Journal of Composite Materials," 2008, doi: 10.1177/0021998308091221.
- [12] A. H. Muliana and S. Sawant, "Responses of viscoelastic polymer composites with temperature and time dependent constituents," Acta Mech., vol. 204, no. 3–4, pp. 155–173, 2009.
- [13] Z. Wang, X. Wang, J. Zhang, W. Liang, and L. Zhou, "Automatic generation of random distribution of fibers in long-fiber- reinforced composites and mesomechanical simulation Automatic generation of random distribution of fibers in long-fiber- reinforced composites and mesomechanical simulation," Mater. Des., vol. 32, no. 2, pp. 885–891, 2011, doi: 10.1016/j.matdes.2010.07.002.
- [14] A. Asadi and J. Raghavan, "Composites : Part B Influence of time-dependent damage on creep of multidirectional polymer composite laminates," Compos. Part B, vol. 42, no. 3, pp. 489–498, 2011, doi: 10.1016/j.compositesb.2010.12.003.
- [15] G. C. Papanicolaou and S. P. Zaoutsos, 1 - Viscoelastic constitutive modeling of creep and stress relaxation in polymers and polymer matrix composites, 2nd ed. Elsevier Ltd., 2019.
- [16] M. F. Sá, A. M. Gomes, J. R. Correia, and N. Silvestre, "Creep behavior of pultruded GFRP elements – Part 1 : Literature review and experimental study," vol. 93, pp. 2450–2459, 2011, doi: 10.1016/j.compstruct.2011.04.013.
- [17] V. Monfared, M. Mondali, and A. Abedian, "Steady-state creep analysis of polymer matrix composites using complex variable method," vol. 227, no. 10, pp. 2182–2194, 2015, doi: 10.1177/0954406212473391.
- [18] L. Ascione, V. P. Berardi, and A. D. Aponte, "Creep phenomena in FRP materials,"

- Mech. Res. Commun., vol. 43, pp. 15–21, 2012, doi: 10.1016/j.mechrescom.2012.03.010.
- [19] M. Romanowicz, “A numerical approach for predicting the failure locus of fiber reinforced composites under combined transverse compression and axial tension,” *Comput. Mater. Sci.*, vol. 51, no. 1, pp. 7–12, 2012, doi: 10.1016/j.commatsci.2011.07.039.
 - [20] T. Publications, “ا 3547,” vol. 811, pp. 135–139, 2013, doi: 10.4028/www.scientific.net/AMR.811.135.
 - [21] A. Sayyidmousavi and H. Bougherara, “The Role of Viscoelasticity on the Fatigue of Angle-ply Polymer Matrix Composites at High and Room Temperatures- A Micromechanical Approach,” 2014, doi: 10.1007/s10443-014-9409-0.
 - [22] R. Ra and B. Mazhari, “International Journal of Mechanical Sciences Modeling creep in polymeric composites : Developing a general integrated procedure,” vol. 99, pp. 112–120, 2015, doi: 10.1016/j.ijmecsci.2015.05.011.
 - [23] “JSTC_Volume 1_Issue 2_Pages 37-42.pdf.” .
 - [24] M. M. Aghdam, S. M. A. Hosseini, and S. R. Morsali, “Simulation of interface damage in metal matrix composites under off-axis loading using cohesive zone model,” *Comput. Mater. Sci.*, vol. 108, pp. 42–47, 2015, doi: 10.1016/j.commatsci.2015.05.030.
 - [25] R. Ansari and M. K. H. Aghdam, “Micromechanics-based viscoelastic analysis of carbon nanotube-reinforced composites subjected to uniaxial and biaxial loading,” *Compos. Part B*, 2015, doi: 10.1016/j.compositesb.2015.10.048.
 - [26] A. Sorzia, “Modelling of Creep and Stress Relaxation Test of a Polypropylene Microfibre by Using Fraction-Exponential Kernel,” vol. 2016, 2016.
 - [27] A. Darvizeh, R. A. Khalkhali, M. J. Mahmoodi, and M. Kazem, “Investigation of interphase effect on the non-linear viscoelastic behavior of multiphase polymer composites,” vol. 16, no. 1, pp. 181–191, 2016.
 - [28] W. Boiler, G. Chaoshun, B. Equipment, and M. Co, “Prediction of Viscoelastic Behavior of Unidirectional Polymer Matrix Composites,” no. June, pp. 695–699, 2016, doi: 10.1007/s11595-016-1431-7.
 - [29] V. Paolo, M. Perrella, L. Feo, and G. Cricri, “Creep behavior of GFRP laminates and their phases : Experimental investigation and analytical modeling,” *Compos. Part B*, vol. 122, pp. 136–144, 2017, doi: 10.1016/j.compositesb.2017.04.015.
 - [30] I. Kaleel, M. Petrolo, A. M. Waas, and E. Carrera, “Computationally efficient, high-fidelity micromechanics framework using refined 1D models,” *Compos. Struct.*, vol. 181, pp. 358–367, 2017.
 - [31] F. Ye and H. Wang, “A simple Python code for computing effective properties of 2D and 3D representative volume element under periodic boundary conditions,” *arXiv Prepr. arXiv1703.03930*, 2017.
 - [32] Z. Li and S. Ghosh, “Micromechanics modeling and validation of thermal-mechanical damage in DER353 epoxy/borosilicate glass composite subject to high strain rate deformation,” *Int. J. Impact Eng.*, vol. 136, p. 103414, 2020.
 - [33] M. Naderi and N. Iyyer, “Micromechanical analysis of damage mechanisms under tension of 0°–90° thin-ply composite laminates,” *Compos. Struct.*, vol. 234, p. 111659, 2020.
 - [34] S. Seyedalikhani, M. M. Shokrieh, and A. R. Shamaei-Kashani, “A novel dynamic constitutive micromechanical model to predict the strain rate dependent mechanical behavior of glass/epoxy laminated composites,” *Polym. Test.*, vol. 82, p. 106292, 2020.

- [35] U. Santhosh, J. Ahmad, G. Ojard, I. Smyth, Y. Gawayed, and G. Jefferson, "Effect of porosity on the nonlinear and time-dependent behavior of Ceramic Matrix Composites," *Compos. Part B*, vol. 184, no. April 2019, p. 107658, 2020, doi: 10.1016/j.compositesb.2019.107658.
- [36] M. A. Eltaher and A. Wagih, "Micromechanical modeling of damage in elasto-plastic nanocomposites using unit cell representative volume element and cohesive zone model," *Ceram. Int.*, 2020.
- [37] H.-B. Huang and Z.-M. Huang, "Micromechanical prediction of elastic-plastic behavior of a short fiber or particle reinforced composite," *Compos. Part A Appl. Sci. Manuf.*, p. 105889, 2020.
- [38] S. S. Godara and P. K. Mahato, "A study on micromechanical methods for the analysis of composite materials," *Mater. Today Proc.*, 2020.
- [39] X. Lou, X. Han, and H. Cai, "A micromechanics-based damage model for compressive behavior analysis of impacted composite laminates," *Int. J. Damage Mech.*, vol. 29, no. 3, pp. 369–387, 2020.
- [40] G. Monastyrckis, L. Mishnaevsky Jr, C. B. Hatter, A. Aniskevich, Y. Gogotsi, and D. Zeleniakiene, "Micromechanical modeling of MXene-polymer composites," *Carbon N. Y.*, 2020.
- [41] F. Nerilli, S. Marfia, and E. Sacco, "Micromechanical modeling of the constitutive response of FRCM composites," *Constr. Build. Mater.*, vol. 236, p. 117539, 2020.
- [42] B. Xu, W. Xu, and F. Guo, "Creep behavior due to interface diffusion in unidirectional fiber-reinforced metal matrix composites under general loading conditions: a micromechanics analysis," *Acta Mech.*, pp. 1–15, 2020.
- [43] R. N. Yancey and M. Pindera, "Micromechanical Analysis of the Creep Response of Unidirectional Composites," vol. 112, pp. 157–163, 1990.
- [44] R. Saravanan, *Manufacturing Optimization through Intelligent Techniques* (2006). CRC Press, 2017.
- [45] ح. توزنده جانی, "ارائه مدل پیشرونده چندمقیاسی برای پیش‌بینی عمر خستگی چندلایه‌های کامپوزیتی تقویت شده با نانوذرات," *دانشگاه صنعتی شریف*, ۱۳۹۵.

پیوست‌ها

پیوست الف) اسکریپت چیست

زبان پایه نرم افزار آباکوس، زبان برنامه نویسی پایتون است. از جمله رایج ترین کارهایی که به کمک اسکریپت نویسی در آباکوس می توان انجام داد، پارامتری کردن مدل های مورد نظر است. به طور کلی پارامتری کردن در نرم افزار آباکوس این امکان را می دهد که بدون شبیه سازی مجدد یک مدل، پارامترهای مورد نظر خود را از قبیل ابعاد هندسی، متریال به کار رفته، نوع مش، مقدار بارگذاری و بسیاری از عوامل دیگر را تغییر داد. به کمک پارامتری کردن مدل در نرم افزار آباکوس قادر خواهند بود کدهای دستوری موجود را در یک حلقه قرار داده و با اعمال الگوریتم های بهینه سازی، اقدام به بهینه کردن مدل کنند. به طور کلی پارامتری کردن مدل در نرم افزار آباکوس سبب می شود که سرعت شبیه سازی بیشتر شده و علاوه بر آن کیفیت تحلیل ثابت باقی بماند.

فایلی که حاوی دستورات محیط اسکریپت نویسی آباکوس است یک اسکریپت نامیده می شود. می توان از یک اسکریپت در موارد زیر استفاده کرد.

- **انجام یک عمل یا دستور تکراری** برای مثال می توانید اسکریپتی ایجاد کرد که به محض آغاز Abaqus/CAE کتابخانه کاملی از مواد مهندسی مورد نظر و پرکاربرد را ایجاد کند تا با ورود به ماژول Property در اختیار شما قرارداد. همچنین می تواند اسکریپتی نوشت که در ماژول Job به اجرای حل بپردازد.
- **ایجاد یک مسئله پارامتریک** برای مثال، می تواند اسکریپتی نوشت که به شکل گام به گام به اصلاح مدل هندسی پرداخته و نتایج تحلیل را آنالیز کند. این اسکریپت می تواند به نحوی نوشته شود تا به اطلاعات خروجی دسترسی داشته، نتایج مورد نظر را نمایش دهد و داده های خاص مورد نظر کاربر را نیز ثبت نماید.

- می‌توانید از طریق اسکریپت‌نویسی به ایجاد و اصلاح مدل‌هایی بپردازد که به شکل مستقیم در Abaqus/CAE در حال ایجاد کردن آن می‌باشید. در واقع رابط کاربری اسکریپت‌نویسی در آباکوس، یک 'API یا برای مدل‌های ایجاد شده توسط کاربر است.
- **دسترسی به داده‌های خروجی** برای مثال، ممکن است نیاز باشد تا به کمک نتایج تحلیل، پس‌پردازش دلخواه ایجاد شود. می‌توان داده‌های دلخواه را در Database Output ثبت کرده و از طریق ماژول Visualization به مشاهده نتایج پرداخت.

¹ Application Programming Interface

پیوست ب) حالت تک فایبر

```

from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import sys
from abaqus import *
from abaqusConstants import *
from odbAccess import *
import xyPlot
import displayGroupOdbToolset as dgo
import os
from types import IntType
import numpy as np
#os.chdir(r"G:\UNIVERSITY\EBRAHIMI THESIS\ABAQUS RUN\RANDOM RVE_V04")
##### CODE INPUT
#####
# Parameter Defination
Lx=1.0
Ly=1.0
Lz=1.0
a=Lx/2.0
b=Ly/2.0
c=Lz/2.0
tol=0.001
ESize=0.1
delta=ESize
Ivf=0
#EEnergy=[0,0,0,0,0,0]
#OUTPUT01 = open("OUTPUT01.plt", "w")
for Vf in [0.1,0.3,0.5,0.7]:
    Ivf=Ivf+1
    MODELNAME='M-'+str(Ivf)
    print('Start Create Model : '+MODELNAME+'\n')
    R=sqrt(Vf*Ly*Lz/pi)
    execfile('UNITCELL_PBC_SQ.py')
    for IBC in [1,2,4,6]:
        print(IBC)
        execfile('UNITCELL_PBC_RUN.py')
        execfile('UNITCELL_PBC_RESULT.py')
        # EEnergy[IBC-1]=StrainEnergy
    #
#####
# ***** Calculation of Effective Modulud of Composite Material
*****

```

```

#
*****
# A1= EEnergy[0]
# A2= EEnergy[1]
# A3= EEnergy[2]
# A4= EEnergy[3]
# A5= EEnergy[4]
# A6= EEnergy[5]
# E11=1.0**2/(2.0*A1)*(Lx*Ly*Lz)
# E22=1.0**2/(2.0*A2)*(Lx*Ly*Lz)
# E33=1.0**2/(2.0*A3)*(Lx*Ly*Lz)
# G12=1.0**2/(2.0*A4)*(Lx*Ly*Lz)
# G13=1.0**2/(2.0*A5)*(Lx*Ly*Lz)
# G23=1.0**2/(2.0*A6)*(Lx*Ly*Lz)
# OUTPUT01.write('{0:10.2f} {1:10.2f} {2:10.2f} {3:10.2f} {4:10.2f}
{5:10.2f} {6:10.2f} {7:10.2f} {8:10.2f} {9:10.2f} {10:10.2f}
\n'.format(Vf,Em,Num,Ef,Nuf,E11,E22,E33,G12,G13,G23))
# OUTPUT01.close()

```

```

#####
from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import sys
from abaqus import *
from abaqusConstants import *
from odbAccess import *
import xyPlot
import displayGroupOdbToolset as dgo
# Lx=1.0
# Ly=1.0
# Lz=1.0
ny=1
nz=1
# tol=0.0005
# a=Lx/2.0
# b=ny*Ly/2.0
# c=nz*Lz/2.0
# ESize=0.1
# delta=ESize
# vf=0.6

# MODELNAME='Model-'+str(j)
mdb.Model(modelType=STANDARD_EXPLICIT, name=MODELNAME)

```

```

mdb.models[MODELNAME].ConstrainedSketch(name='__profile__', sheetSize=Lx)
mdb.models[MODELNAME].sketches['__profile__'].rectangle(point1=(-Lx/2, -
Ly/2),
    point2=(Lx/2, Ly/2))
mdb.models[MODELNAME].Part(dimensionality=THREE_D, name='Part-1', type=
DEFORMABLE_BODY)
mdb.models[MODELNAME].parts['Part-1'].BaseSolidExtrude(depth=Lz/2, sketch=
mdb.models[MODELNAME].sketches['__profile__'])
mdb.models[MODELNAME].parts['Part-1'].Mirror(keepOriginal=ON, mirrorPlane=
mdb.models[MODELNAME].parts['Part-1'].faces[5])
mdb.models[MODELNAME].ConstrainedSketch(gridSpacing=0.07,
name='__profile__',
    sheetSize=3.0, transform=
    mdb.models[MODELNAME].parts['Part-1'].MakeSketchTransform(
    sketchPlane=mdb.models[MODELNAME].parts['Part-1'].faces[1],
    sketchPlaneSide=SIDE1,
    sketchUpEdge=mdb.models[MODELNAME].parts['Part-1'].edges[7],
    sketchOrientation=RIGHT, origin=(-0.5, 0.0, 0.0)))
mdb.models[MODELNAME].parts['Part-1'].projectReferencesOntoSketch(filter=
COPLANAR_EDGES,
sketch=mdb.models[MODELNAME].sketches['__profile__'])
mdb.models[MODELNAME].sketches['__profile__'].CircleByCenterPerimeter(cente
r=(
    0.0, 0.0), point1=(R, 0.0))
mdb.models[MODELNAME].sketches['__profile__'].rectangle(point1=(-R/2, -
R/2),
    point2=(R/2, R/2))
mdb.models[MODELNAME].parts['Part-1'].PartitionCellBySketch(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx, -
Ly, -Lz, Lx, Ly, Lz)
    , sketch=mdb.models[MODELNAME].sketches['__profile__'],
sketchPlane=
    mdb.models[MODELNAME].parts['Part-1'].faces[1], sketchUpEdge=
    mdb.models[MODELNAME].parts['Part-1'].edges[7])
del mdb.models[MODELNAME].sketches['__profile__']
mdb.models[MODELNAME].parts['Part-1'].PartitionCellByExtrudeEdge(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx, -
Ly, -Lz, Lx, Ly, Lz)
    , edges=(mdb.models[MODELNAME].parts['Part-1'].edges[0], ), line=
    mdb.models[MODELNAME].parts['Part-1'].edges[11], sense=FORWARD)
mdb.models[MODELNAME].parts['Part-1'].PartitionCellByExtrudeEdge(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx, -
Ly, -Lz, Lx, Ly, Lz)
    , edges=(mdb.models[MODELNAME].parts['Part-1'].edges[8],
    mdb.models[MODELNAME].parts['Part-1'].edges[9],
    mdb.models[MODELNAME].parts['Part-1'].edges[10],
    mdb.models[MODELNAME].parts['Part-1'].edges[11]), line=
    mdb.models[MODELNAME].parts['Part-1'].edges[12], sense=FORWARD)
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=mdb.models[MODELNAME].parts['Part
-1'].cells[0:2],
    point1=mdb.models[MODELNAME].parts['Part-1'].vertices[11], point2=
    mdb.models[MODELNAME].parts['Part-1'].vertices[14], point3=
    mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
    mdb.models[MODELNAME].parts['Part-1'].edges[22], MIDDLE))
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=mdb.models[MODELNAME].parts['Part
-1'].cells[0:4],
    point1=mdb.models[MODELNAME].parts['Part-1'].vertices[16], point2=

```

```

        mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
            mdb.models[MODELNAME].parts['Part-1'].edges[39], MIDDLE), point3=
            mdb.models[MODELNAME].parts['Part-1'].vertices[23])
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz),
    point1=mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
        mdb.models[MODELNAME].parts['Part-1'].edges[16], MIDDLE), point2=
        mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
            mdb.models[MODELNAME].parts['Part-1'].edges[37], MIDDLE), point3=
            mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
                mdb.models[MODELNAME].parts['Part-1'].edges[50], MIDDLE))
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz),
    point1=mdb.models[MODELNAME].parts['Part-1'].vertices[41],
    point2=mdb.models[MODELNAME].parts['Part-1'].vertices[40],
    point3=mdb.models[MODELNAME].parts['Part-1'].vertices[35])

mdb.models[MODELNAME].Material(name='Epoxy')
mdb.models[MODELNAME].materials['Epoxy'].Creep(dependencies=0, law=STRAIN,
    table=((1.10E-06, 1.0, -0.8218, 22.0), ), temperatureDependency=ON)
mdb.models[MODELNAME].materials['Epoxy'].setValues(materialIdentifier='')
mdb.models[MODELNAME].materials['Epoxy'].setValues(description='')
mdb.models[MODELNAME].materials['Epoxy'].Elastic(dependencies=0, moduli=
    LONG_TERM, noCompression=OFF, noTension=OFF, table=((4510.0, 0.311), ),
    temperatureDependency=OFF, type=ISOTROPIC)
mdb.models[MODELNAME].Material(name='Carbon Fiber')
mdb.models[MODELNAME].materials['Carbon
Fiber'].setValues(materialIdentifier='')
mdb.models[MODELNAME].materials['Carbon Fiber'].setValues(description='')
mdb.models[MODELNAME].materials['Carbon Fiber'].Elastic(dependencies=0,
moduli=
    LONG_TERM, noCompression=OFF, noTension=OFF, table=((202820.0, 25300.0,
25300.0, 0.443, 0.443, 0.0, 44120.0, 44120.0, 44120.0), ),
    temperatureDependency=OFF, type=ENGINEERING_CONSTANTS)

mdb.models[MODELNAME].HomogeneousSolidSection(material='Carbon Fiber',
name=
    'FIBER', thickness=None)
mdb.models[MODELNAME].HomogeneousSolidSection(material='Epoxy',
name='MATRIX',
    thickness=None)
FIBER=mdb.models[MODELNAME].parts['Part-1'].Set(cells=
    mdb.models[MODELNAME].parts['Part-
1'].cells.getByBoundingCylinder((-Lx,0.0,0.0), (Lx,0.0,0.0),R+tol),
name='FIBER')
COMPOSITE=mdb.models[MODELNAME].parts['Part-1'].Set(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz), name='COMPOSITE')
MATRIX=mdb.models[MODELNAME].parts['Part-
1'].SetByBoolean(name='MATRIX',sets=(COMPOSITE,FIBER),operation=DIFFERENCE)

mdb.models[MODELNAME].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
        mdb.models[MODELNAME].parts['Part-1'].sets['FIBER'],
sectionName='FIBER',

```

```

        thicknessAssignment=FROM_SECTION)
# mdb.models[MODELNAME].parts['Part-1'].Set(cells=
#   # mdb.models[MODELNAME].parts['Part-1'].cells.getSequenceFromMask((
#   # '[#4ce06 ]', ), ), name='MATRIX')
mdb.models[MODELNAME].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models[MODELNAME].parts['Part-1'].sets['MATRIX'],
sectionName='MATRIX',
    thicknessAssignment=FROM_SECTION)

mdb.models[MODELNAME].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models[MODELNAME].rootAssembly.Instance(dependent=ON, name='Part-1-1',
    part=mdb.models[MODELNAME].parts['Part-1'])
mdb.models[MODELNAME].rootAssembly.translate(instanceList=('Part-1-1', ),
    vector=(0.0, (1-ny)*Ly/2, (1-nz)*Lz/2))
if (ny>1 or nz>1):
    mdb.models[MODELNAME].rootAssembly.LinearInstancePattern(direction1
=(0.0, 1.0,
        0.0), direction2=(0.0, 0.0, 1.0), instanceList=('Part-1-1',
), number1=ny,
        number2=nz, spacing1=1.0, spacing2=1.0)
    my_list = []
    RA=mdb.models[MODELNAME].rootAssembly
    for yi in range(ny):
        for zi in range(nz):
            if (yi==0 and zi==0):
                name = RA.instances[str('Part-1-1')]
            else:
                name = RA.instances[str('Part-1-1-lin-
')+str(yi+1)+str('-')+str(zi+1)]
                my_list.append(name)
    my_list = tuple(my_list)
    mdb.models[MODELNAME].rootAssembly.InstanceFromBooleanMerge(domain=
BOTH,
        instances=my_list, keepIntersections=ON,
mergeNodes=BOUNDARY_ONLY, name='Part-2',
        originalInstances=DELETE)
    mdb.models[MODELNAME].parts['Part-
2'].seedEdgeBySize(constraint=FINER,
        deviationFactor=ESize, edges=
        mdb.models[MODELNAME].parts['Part-
2'].edges.getByBoundingBox(-100, -100, -100, 100,100, 100), size=ESize)
    mdb.models[MODELNAME].parts['Part-2'].generateMesh()
    PName=str('Part-2')
else:
    mdb.models[MODELNAME].parts['Part-
1'].seedEdgeBySize(constraint=FINER,
        deviationFactor=ESize, edges=
        mdb.models[MODELNAME].parts['Part-
1'].edges.getByBoundingBox(-100, -100, -100, 100,100, 100), size=ESize)
    mdb.models[MODELNAME].parts['Part-1'].generateMesh()
    PName=str('Part-1')
mdb.models[MODELNAME].rootAssembly.regenerate()

mdb.models[MODELNAME].ViscoStep(cetol=0.001, initialInc=72.0,
maxInc=7200.0,
    maxNumInc=1000000, minInc=0.072, name='Step-1', nlgeom=ON, previous=

```

```

        'Initial', timePeriod=7200.0)
mdb.models[MODELNAME].TimePoint(name='TimePoints-1', points=((0.0, 100.0,
20.0), (
    100.0, 1000.0, 100.0), (1000.0, 10000.0, 500.0)))
mdb.models[MODELNAME].fieldOutputRequests['F-Output-
1'].setValues(timePoint=
    'TimePoints-1', variables=('S', 'E', 'PE', 'PEEQ', 'PEMAG', 'EE', 'CE',
    'CEP', 'CEEQ', 'LE', 'U', 'RF', 'CF', 'CSTRESS', 'CDISP', 'ENER',
    'ELEN',
    'ELEDEN', 'EVOL'))

mdb.models[MODELNAME].parts[PName].MaterialOrientation(
    additionalRotationType=ROTATION_NONE, axis=AXIS_1, fieldName='',
localCsys=
    None, orientationType=GLOBAL, region=
    mdb.models[MODELNAME].parts[PName].sets['FIBER'], stackDirection=
    STACK_3)
mdb.models[MODELNAME].parts[PName].MaterialOrientation(
    additionalRotationType=ROTATION_NONE, axis=AXIS_1, fieldName='',
localCsys=
    None, orientationType=GLOBAL, region=
    mdb.models[MODELNAME].parts[PName].sets['MATRIX'], stackDirection=
    STACK_3)

#*****
#*****
# ***** Create Reference Point Set
#*****
#
#*****
#*****
RP_E11=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E22=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E33=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E12=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E13=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E23=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))

mdb.models[MODELNAME].rootAssembly.Set(name='RP_E11', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E11.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E22', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E22.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E33', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E33.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E12', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E12.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E13', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E13.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E23', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E23.id], ))

```

```

#*****
#*****
# ***** Create Boundry Condition Node Set
#*****
#
#*****
# ***** Ceaate Set for node on the surface but not on the edges and
corner *****
XN=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,-c+tol,-a+tol,b+tol,c+tol)
XP=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a+tol,-b+tol,-c+tol,a+tol,b+tol,c+tol)
YN=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,-c+tol,a+tol,-b+tol,c+tol)
YP=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,b+tol,-c+tol,a+tol,b+tol,c+tol)
ZN=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,-c+tol,a+tol,b+tol,-c+tol)
ZP=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,c+tol,a+tol,b+tol,c+tol)
mdb.models[MODELNAME].rootAssembly.Set(name='XN', nodes=XN)
mdb.models[MODELNAME].rootAssembly.Set(name='XP', nodes=XP)
mdb.models[MODELNAME].rootAssembly.Set(name='YN', nodes=YN)
mdb.models[MODELNAME].rootAssembly.Set(name='YP', nodes=YP)
mdb.models[MODELNAME].rootAssembly.Set(name='ZN', nodes=ZN)
mdb.models[MODELNAME].rootAssembly.Set(name='ZP', nodes=ZP)
for IXN in range (len(XN)):
    for IXP in range (len(XP)):
        if (abs(XN[IXN].coordinates[1]-
XP[IXP].coordinates[1])<tol) and (abs(XN[IXN].coordinates[2]-
XP[IXP].coordinates[2])<tol):
            mdb.models[MODELNAME].rootAssembly.Set(name='XN-
'+str(IXN+1), nodes=XN[IXN:IXN+1])
            mdb.models[MODELNAME].rootAssembly.Set(name='XP-
'+str(IXN+1), nodes= XP[IXP:IXP+1])
for IYN in range (len(YN)):
    for IYP in range (len(YP)):
        if (abs(YN[IYN].coordinates[0]-
YP[IYP].coordinates[0])<tol) and (abs(YN[IYN].coordinates[2]-
YP[IYP].coordinates[2])<tol):
            mdb.models[MODELNAME].rootAssembly.Set(name='YN-
'+str(IYN+1), nodes=YN[IYN:IYN+1])
            mdb.models[MODELNAME].rootAssembly.Set(name='YP-
'+str(IYN+1), nodes= YP[IYP:IYP+1])
for IZN in range (len(ZN)):
    for IZP in range (len(ZP)):
        if (abs(ZN[IZN].coordinates[0]-
ZP[IZP].coordinates[0])<tol) and (abs(ZN[IZN].coordinates[1]-
ZP[IZP].coordinates[1])<tol):
            mdb.models[MODELNAME].rootAssembly.Set(name='ZN-
'+str(IZN+1), nodes=ZN[IZN:IZN+1])
            mdb.models[MODELNAME].rootAssembly.Set(name='ZP-
'+str(IZN+1), nodes= ZP[IZP:IZP+1])

for IX in range(len(XN)):
    mdb.models[MODELNAME].Equation(name='SCEXX-'+str(IX+1),
terms=((-1.0, 'XP-'+str(IX+1),1),
(1.0, 'XN-'+str(IX+1), 1),(1, 'RP_E11', 1)))

```

```

        mdb.models[MODELNAME].Equation(name='SCEXY-'+str(IX+1),
terms=((-1.0, 'XP-'+str(IX+1), 2),
        (1.0, 'XN-'+str(IX+1), 2), (1, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='SCEXZ-'+str(IX+1),
terms=((-1.0, 'XP-'+str(IX+1), 3),
        (1.0, 'XN-'+str(IX+1), 3), (1, 'RP_E13', 1)))
for IY in range(len(YN)):
        mdb.models[MODELNAME].Equation(name='SCEYX-'+str(IY+1),
terms=((-1.0, 'YP-'+str(IY+1), 1),
        (1.0, 'YN-'+str(IY+1), 1), (b/a, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='SCEYY-'+str(IY+1),
terms=((-1.0, 'YP-'+str(IY+1), 2),
        (1.0, 'YN-'+str(IY+1), 2), (1, 'RP_E22', 1)))
        mdb.models[MODELNAME].Equation(name='SCEYZ-'+str(IY+1),
terms=((-1.0, 'YP-'+str(IY+1), 3),
        (1.0, 'YN-'+str(IY+1), 3), (1, 'RP_E23', 1)))
for IZ in range(len(ZN)):
        mdb.models[MODELNAME].Equation(name='SCEZX-'+str(IZ+1),
terms=((-1.0, 'ZP-'+str(IZ+1), 1),
        (1.0, 'ZN-'+str(IZ+1), 1), (c/a, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='SCEZY-'+str(IZ+1),
terms=((-1.0, 'ZP-'+str(IZ+1), 2),
        (1.0, 'ZN-'+str(IZ+1), 2), (c/b, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='SCEZZ-'+str(IZ+1),
terms=((-1.0, 'ZP-'+str(IZ+1), 3),
        (1.0, 'ZN-'+str(IZ+1), 3), (1, 'RP_E33', 1)))

#
*****
*****
# ***** Ceaaate Set for node  on the edges but not on
corner *****
XYNN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b-tol,-c+tol,-a+tol,-b+tol,c-tol)
XYNP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,b-tol,-c+tol,-a+tol,b+tol,c-tol)
XZNN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b+tol,-c-tol,-a+tol,b-tol,-c+tol)
XZNP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b+tol,c-tol,-a+tol,b-tol,c+tol)
XYPN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b-tol,-c+tol,a+tol,-b+tol,c-tol)
XYPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,b-tol,-c+tol,a+tol,b+tol,c-tol)
XZPN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b+tol,-c-tol,a+tol,b-tol,-c+tol)
XZPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b+tol,c-tol,a+tol,b-tol,c+tol)
YZNN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b-tol,-c-tol,a-tol,-b+tol,-c+tol)
YZNP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b-tol,c-tol,a-tol,-b+tol,c+tol)
YZPN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,b-tol,-c-tol,a-tol,b+tol,-c+tol)
YZPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,b-tol,c-tol,a-tol,b+tol,c+tol)
mdb.models[MODELNAME].rootAssembly.Set(name='XYNN', nodes=XYNN)
mdb.models[MODELNAME].rootAssembly.Set(name='XYNP', nodes=XYNP)
mdb.models[MODELNAME].rootAssembly.Set(name='XZNN', nodes=XZNN)
mdb.models[MODELNAME].rootAssembly.Set(name='XZNP', nodes=XZNP)

```

```

mdb.models[MODELNAME].rootAssembly.Set(name='XYPN', nodes=XYPN)
mdb.models[MODELNAME].rootAssembly.Set(name='XYPP', nodes=XYPP)
mdb.models[MODELNAME].rootAssembly.Set(name='XZPN', nodes=XZPN)
mdb.models[MODELNAME].rootAssembly.Set(name='XZPP', nodes=XZPP)
mdb.models[MODELNAME].rootAssembly.Set(name='YZNN', nodes=YZNN)
mdb.models[MODELNAME].rootAssembly.Set(name='YZNP', nodes=YZNP)
mdb.models[MODELNAME].rootAssembly.Set(name='YZPN', nodes=YZPN)
mdb.models[MODELNAME].rootAssembly.Set(name='YZPP', nodes=YZPP)

for IXYNN in range(len(XYNN)):
    for IXYNP in range(len(XYNP)):
        if (abs(XYNN[IXYNN].coordinates[2]-
XYNP[IXYNP].coordinates[2])<tol):
            for IXYPN in range(len(XYPN)):
                if (abs(XYNN[IXYNN].coordinates[2]-
XYPN[IXYPN].coordinates[2])<tol):
                    for IXYPP in range(len(XYPP)):
                        if
(abs(XYNN[IXYNN].coordinates[2]-XYPP[IXYPP].coordinates[2])<tol):

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYNN-'+str(IXYNN+1),
nodes=XYNN[IXYNN:IXYNN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYNP-'+str(IXYNN+1),
nodes=XYNP[IXYNP:IXYNP+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYPN-'+str(IXYNN+1),
nodes=XYPN[IXYPN:IXYPN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYPP-'+str(IXYNN+1),
nodes=XYPP[IXYPP:IXYPP+1])
for IXZNN in range(len(XZNN)):
    for IXZNP in range(len(XZNP)):
        if (abs(XZNN[IXZNN].coordinates[1]-
XZNP[IXZNP].coordinates[1])<tol):
            for IXZPN in range(len(XZPN)):
                if (abs(XZNN[IXZNN].coordinates[1]-
XZPN[IXZPN].coordinates[1])<tol):
                    for IXZPP in range(len(XZPP)):
                        if
(abs(XZNN[IXZNN].coordinates[1]-XZPP[IXZPP].coordinates[1])<tol):

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZNN-'+str(IXZNN+1),
nodes=XZNN[IXZNN:IXZNN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZNP-'+str(IXZNN+1),
nodes=XZNP[IXZNP:IXZNP+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZPN-'+str(IXZNN+1),
nodes=XZPN[IXZPN:IXZPN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZPP-'+str(IXZNN+1),
nodes=XZPP[IXZPP:IXZPP+1])
for IYZNN in range(len(YZNN)):
    for IYZNP in range(len(YZNP)):
        if (abs(YZNN[IYZNN].coordinates[0]-
YZNP[IYZNP].coordinates[0])<tol):
            for IYZPN in range(len(YZPN)):

```

```

        if (abs(YZNN[IYZNN].coordinates[0]-
YZPN[IYZPN].coordinates[0])<tol):
            for IYZPP in range (len(YZPP)):
                if
(abs(YZNN[IYZNN].coordinates[0]-YZPP[IYZPP].coordinates[0])<tol):

                    mdb.models[MODELNAME].rootAssembly.Set(name='YZNN-'+str(IYZNN+1),
nodes=YZNN[IYZNN:IYZNN+1])

                    mdb.models[MODELNAME].rootAssembly.Set(name='YZNP-'+str(IYZNN+1),
nodes=YZNP[IYZNP:IYZNP+1])

                    mdb.models[MODELNAME].rootAssembly.Set(name='YZPN-'+str(IYZNN+1),
nodes=YZPN[IYZPN:IYZPN+1])

                    mdb.models[MODELNAME].rootAssembly.Set(name='YZPP-'+str(IYZNN+1),
nodes=YZPP[IYZPP:IYZPP+1])

for IXY in range(len(XYNN)):
    mdb.models[MODELNAME].Equation(name='ECXY_X1-'+str(IXY+1),
terms=((-1.0, 'XYPP-'+str(IXY+1),1),
        (1.0, 'XYPN-'+str(IXY+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_X2-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),1),
        (1.0, 'XYNN-'+str(IXY+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_X3-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),1),
        (1.0, 'XYNN-'+str(IXY+1), 1),(b/a, 'RP_E12', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Y1-'+str(IXY+1),
terms=((-1.0, 'XYPP-'+str(IXY+1),2),
        (1.0, 'XYPN-'+str(IXY+1), 2),(1, 'RP_E12', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Y2-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),2),
        (1.0, 'XYNN-'+str(IXY+1), 2),(1, 'RP_E12', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Y3-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),2),
        (1.0, 'XYNN-'+str(IXY+1), 2),(1, 'RP_E22', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Z1-'+str(IXY+1),
terms=((-1.0, 'XYPP-'+str(IXY+1),3),
        (1.0, 'XYPN-'+str(IXY+1), 3),(1, 'RP_E13', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Z2-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),3),
        (1.0, 'XYNN-'+str(IXY+1), 3),(1, 'RP_E13', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Z3-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),3),
        (1.0, 'XYNN-'+str(IXY+1), 3),(1, 'RP_E23', 1)))
for IXZ in range(len(XZNN)):
    mdb.models[MODELNAME].Equation(name='ECXZ_X1-'+str(IXZ+1),
terms=((-1.0, 'XZPP-'+str(IXZ+1),1),
        (1.0, 'XZNP-'+str(IXZ+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXZ_X2-'+str(IXZ+1),
terms=((-1.0, 'XZPN-'+str(IXZ+1),1),
        (1.0, 'XZNN-'+str(IXZ+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXZ_X3-'+str(IXZ+1),
terms=((-1.0, 'XZNP-'+str(IXZ+1),1),
        (1.0, 'XZNN-'+str(IXZ+1), 1),(c/a, 'RP_E13', 1)))
    mdb.models[MODELNAME].Equation(name='ECXZ_Y1-'+str(IXZ+1),
terms=((-1.0, 'XZPP-'+str(IXZ+1),2),
        (1.0, 'XZNP-'+str(IXZ+1), 2),(1, 'RP_E12', 1)))

```

```

        mdb.models[MODELNAME].Equation(name='ECXZ_Y2-'+str(IXZ+1),
terms=((-1.0, 'XZPN-'+str(IXZ+1), 2),
        (1.0, 'XZNN-'+str(IXZ+1), 2), (1, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Y3-'+str(IXZ+1),
terms=((-1.0, 'XZNP-'+str(IXZ+1), 2),
        (1.0, 'XZNN-'+str(IXZ+1), 2), (c/b, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Z1-'+str(IXZ+1),
terms=((-1.0, 'XZPP-'+str(IXZ+1), 3),
        (1.0, 'XZNP-'+str(IXZ+1), 3), (1, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Z2-'+str(IXZ+1),
terms=((-1.0, 'XZPN-'+str(IXZ+1), 3),
        (1.0, 'XZNN-'+str(IXZ+1), 3), (1, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Z3-'+str(IXZ+1),
terms=((-1.0, 'XZNP-'+str(IXZ+1), 3),
        (1.0, 'XZNN-'+str(IXZ+1), 3), (1, 'RP_E33', 1)))
for IYZ in range(len(YZNN)):
        mdb.models[MODELNAME].Equation(name='ECYZ_X1-'+str(IYZ+1),
terms=((-1.0, 'YZPP-'+str(IYZ+1), 1),
        (1.0, 'YZNP-'+str(IYZ+1), 1), (b/a, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_X2-'+str(IYZ+1),
terms=((-1.0, 'YZPN-'+str(IYZ+1), 1),
        (1.0, 'YZNN-'+str(IYZ+1), 1), (b/a, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_X3-'+str(IYZ+1),
terms=((-1.0, 'YZNP-'+str(IYZ+1), 1),
        (1.0, 'YZNN-'+str(IYZ+1), 1), (c/a, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Y1-'+str(IYZ+1),
terms=((-1.0, 'YZPP-'+str(IYZ+1), 2),
        (1.0, 'YZNP-'+str(IYZ+1), 2), (1, 'RP_E22', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Y2-'+str(IYZ+1),
terms=((-1.0, 'YZPN-'+str(IYZ+1), 2),
        (1.0, 'YZNN-'+str(IYZ+1), 2), (1, 'RP_E22', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Y3-'+str(IYZ+1),
terms=((-1.0, 'YZNP-'+str(IYZ+1), 2),
        (1.0, 'YZNN-'+str(IYZ+1), 2), (c/b, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Z1-'+str(IYZ+1),
terms=((-1.0, 'YZPP-'+str(IYZ+1), 3),
        (1.0, 'YZNP-'+str(IYZ+1), 3), (1, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Z2-'+str(IYZ+1),
terms=((-1.0, 'YZPN-'+str(IYZ+1), 3),
        (1.0, 'YZNN-'+str(IYZ+1), 3), (1, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Z3-'+str(IYZ+1),
terms=((-1.0, 'YZNP-'+str(IYZ+1), 3),
        (1.0, 'YZNN-'+str(IYZ+1), 3), (1, 'RP_E33', 1)))
#
*****
*****
# ***** Ceaaate Set for node on the corner
*****
mdb.models[MODELNAME].rootAssembly.Set(name='NNN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')] .nodes.getByBoundingBox(-a-tol, -b-tol, -c-tol, -a+tol, -b+tol, -c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='NPN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')] .nodes.getByBoundingBox(-a-tol, b-tol, -c-tol, -a+tol, b+tol, -c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='NPP', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')] .nodes.getByBoundingBox(-a-tol, b-tol, c-tol, -a+tol, b+tol, c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='NNP', nodes=

```

```

        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b-tol,c-tol,-a+tol,-b+tol,c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PNN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b-tol,-c-tol,a+tol,-b+tol,-c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PPN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,b-tol,-c-tol,a+tol,b+tol,-c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PPP', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,b-tol,c-tol,a+tol,b+tol,c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PNP', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b-tol,c-tol,a+tol,-b+tol,c+tol))

mdb.models[MODELNAME].Equation(name='CX1', terms=((-1.0, 'PPP', 1),
(1.0, 'NPP', 1), (1, 'RP_E11', 1)))
mdb.models[MODELNAME].Equation(name='CX2', terms=((-1.0, 'PNP', 1),
(1.0, 'NNP', 1), (1, 'RP_E11', 1)))
mdb.models[MODELNAME].Equation(name='CX3', terms=((-1.0, 'PNN', 1),
(1.0, 'NNN', 1), (1, 'RP_E11', 1)))
mdb.models[MODELNAME].Equation(name='CX4', terms=((-1.0, 'PPN', 1),
(1.0, 'PNN', 1), (b/a, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CX5', terms=((-1.0, 'NPN', 1),
(1.0, 'NNN', 1), (b/a, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CX6', terms=((-1.0, 'NPP', 1),
(1.0, 'NPN', 1), (c/a, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CX7', terms=((-1.0, 'NNP', 1),
(1.0, 'NNN', 1), (c/a, 'RP_E13', 1)))

mdb.models[MODELNAME].Equation(name='CY1', terms=((-1.0, 'PPP', 2),
(1.0, 'NPP', 2), (1, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CY2', terms=((-1.0, 'PNP', 2),
(1.0, 'NNP', 2), (1, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CY3', terms=((-1.0, 'PNN', 2),
(1.0, 'NNN', 2), (1, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CY4', terms=((-1.0, 'PPN', 2),
(1.0, 'PNN', 2), (1, 'RP_E22', 1)))
mdb.models[MODELNAME].Equation(name='CY5', terms=((-1.0, 'NPN', 2),
(1.0, 'NNN', 2), (1, 'RP_E22', 1)))
mdb.models[MODELNAME].Equation(name='CY6', terms=((-1.0, 'NPP', 2),
(1.0, 'NPN', 2), (c/b, 'RP_E23', 1)))
mdb.models[MODELNAME].Equation(name='CY7', terms=((-1.0, 'NNP', 2),
(1.0, 'NNN', 2), (c/b, 'RP_E23', 1)))

mdb.models[MODELNAME].Equation(name='CZ1', terms=((-1.0, 'PPP', 3),
(1.0, 'NPP', 3), (1, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CZ2', terms=((-1.0, 'PNP', 3),
(1.0, 'NNP', 3), (1, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CZ3', terms=((-1.0, 'PNN', 3),
(1.0, 'NNN', 3), (1, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CZ4', terms=((-1.0, 'PPN', 3),
(1.0, 'PNN', 3), (1, 'RP_E23', 1)))
mdb.models[MODELNAME].Equation(name='CZ5', terms=((-1.0, 'NPN', 3),
(1.0, 'NNN', 3), (1, 'RP_E23', 1)))
mdb.models[MODELNAME].Equation(name='CZ6', terms=((-1.0, 'NPP', 3),
(1.0, 'NPN', 3), (1, 'RP_E33', 1)))
mdb.models[MODELNAME].Equation(name='CZ7', terms=((-1.0, 'NNP', 3),

```

```

(1.0, 'NNN', 3), (1, 'RP_E33', 1)))
#####

if IBC==1:
    BC_SET='RP_E11'
    CF=1.0*b*c**4*1.0

elif IBC==2:
    BC_SET='RP_E22'
    CF=1.0*a*c**4*1.0

elif IBC==3:
    BC_SET='RP_E33'
    CF=1.0*b*a**4*1.0

elif IBC==4:#Sxy
    BC_SET='RP_E12'
    CF=2.0*b*c**4*1.0

elif IBC==5:#Sxz
    BC_SET='RP_E13'
    CF=2.0*b*c**4*1.0

else:#Syz
    BC_SET='RP_E23'
    CF=2.0*a*c**4*1.0

#
*****
****
#***** Boundary Conditions
*****
#
*****
****
mdb.models[MODELNAME].TabularAmplitude(name='Amp-1', timeSpan=STEP,
    smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1.0, 1.0), (7200.0, 1.0)))
mdb.models[MODELNAME].ConcentratedForce(name='Load-1', cfl=CF,
createStepName='Step-1', amplitude='Amp-1',
    distributionType=UNIFORM, field='', localCsys=None,
region=mdb.models[MODELNAME].rootAssembly.sets[BC_SET])
# mdb.models[MODELNAME].ConcentratedForce(cfl=CF, createStepName='Step-1',
    # distributionType=UNIFORM, field='', localCsys=None, name='Load-
1', region=
    # mdb.models[MODELNAME].rootAssembly.sets[BC_SET])
#
*****
****
#***** Job Genration
*****
#
*****
****
Job_Name=MODELNAME+'-'+str(IBC)
mdb.models[MODELNAME].historyOutputRequests['H-Output-
1'].setValues(timePoint=

```

```

        'TimePoints-1')
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
        explicitPrecision=SINGLE, getMemoryFromAnalysis=True,
historyPrint=OFF,
        memory=50, memoryUnits=PERCENTAGE, model=MODELNAME, modelPrint=OFF,
        multiprocessingMode=DEFAULT, name=Job_Name,
nodalOutputPrecision=SINGLE,
        numCpus=2, numDomains=2, queue=None, scratch='', type=ANALYSIS,
        userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs[Job_Name].submit(consistencyChecking=OFF)
mdb.jobs[Job_Name].waitForCompletion()
#####

```

```

odb = session.openOdb(Job_Name+'.odb')

Ese = session.XYDataFromHistory(name='StrainEnergy',
odb=odb,outputVariableName='Strain energy: ALLSE for Whole Model',
steps=('Step-1',), )
#StrainEnergy=Ese[len(xy_result)-1][1]
# SE = session.Curve(xyData=Ese).data
# outputF1 = open( Job_Name + '_Ese.csv', 'wb')
# writer = csv.writer(outputF1)
# writer.writerows(SE)
# outputF1.close()
Cde = session.XYDataFromHistory(name='ALLCD Whole Model-1',
odb=odb,outputVariableName='Creep dissipation energy: ALLCD for Whole
Model',steps=('Step-1', ),)
DAT=(Ese+Cde)*2.0 # 2U=Sigma^2/E
with open(Job_Name + '_Ese.plt', 'wb') as f:
    np.savetxt(f,Ese,delimiter=' ')
with open(Job_Name + '_Cde.plt', 'wb') as f:
    np.savetxt(f,Cde,delimiter=' ')
with open(Job_Name + '_Te.plt', 'wb') as f:
    np.savetxt(f,DAT,delimiter=' ')

```

پیوست ج) حالت فاز میانی

```

from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import sys
from abaqus import *
from abaqusConstants import *
from odbAccess import *
import xyPlot
import displayGroupOdbToolset as dgo
import os
from types import IntType
import numpy as np
#os.chdir(r"G:\UNIVERSITY\EBRAHIMI THESIS\ABAQUS RUN\RANOME RVE_V04")
##### CODE INPUT
#####
# Parameter Defination
Lx=1.0
Ly=1.0
Lz=1.0
a=Lx/2.0
b=Ly/2.0
c=Lz/2.0
tol=0.001
ESize=0.1
delta=ESize
Ivf=0
tc=0.08
#EEnergy=[0,0,0,0,0,0]
#OUTPUT01 = open("OUTPUT01.plt", "w")
for Vf in [0.5]:
    Ivf=Ivf+1
    MODELNAME='M-'+str(Ivf)
    print('Start Create Model : '+MODELNAME+'\n')
    R=sqrt(Vf*Ly*Lz/pi)
    execfile('UNITCELL_PBC_SQ.py')
    for IBC in [1,2,4,6]:
        print(IBC)
        execfile('UNITCELL_PBC_RUN.py')
        execfile('UNITCELL_PBC_RESULT.py')
        # EEnergy[IBC-1]=StrainEnergy
    #
*****
*****
# ***** Calculation of Effective Modulud of Composite Material
*****

```

```

#
*****
# A1= EEnergy[0]
# A2= EEnergy[1]
# A3= EEnergy[2]
# A4= EEnergy[3]
# A5= EEnergy[4]
# A6= EEnergy[5]
# E11=1.0**2/(2.0*A1)*(Lx*Ly*Lz)
# E22=1.0**2/(2.0*A2)*(Lx*Ly*Lz)
# E33=1.0**2/(2.0*A3)*(Lx*Ly*Lz)
# G12=1.0**2/(2.0*A4)*(Lx*Ly*Lz)
# G13=1.0**2/(2.0*A5)*(Lx*Ly*Lz)
# G23=1.0**2/(2.0*A6)*(Lx*Ly*Lz)
# OUTPUT01.write('{0:10.2f} {1:10.2f} {2:10.2f} {3:10.2f} {4:10.2f}
{5:10.2f} {6:10.2f} {7:10.2f} {8:10.2f} {9:10.2f} {10:10.2f}
\n'.format(Vf,Em,Num,Ef,Nuf,E11,E22,E33,G12,G13,G23))
# OUTPUT01.close()
#####

```

```

from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import sys
from abaqus import *
from abaqusConstants import *
from odbAccess import *
import xyPlot
import displayGroupOdbToolset as dgo
# Lx=1.0
# Ly=1.0
# Lz=1.0
ny=1
nz=1
# tol=0.0005
# a=Lx/2.0
# b=ny*Ly/2.0
# c=nz*Lz/2.0
# ESize=0.1
# delta=ESize
# vf=0.6

# MODELNAME='Model-'+str(j)
mdb.Model(modelType=STANDARD_EXPLICIT, name=MODELNAME)

mdb.models[MODELNAME].ConstrainedSketch(name='__profile__', sheetSize=Lx)

```

```

mdb.models[MODELNAME].sketches['__profile__'].rectangle(point1=(-Lx/2, -
Ly/2),
    point2=(Lx/2, Ly/2))
mdb.models[MODELNAME].Part(dimensionality=THREE_D, name='Part-1', type=
DEFORMABLE_BODY)
mdb.models[MODELNAME].parts['Part-1'].BaseSolidExtrude(depth=Lz/2, sketch=
mdb.models[MODELNAME].sketches['__profile__'])
mdb.models[MODELNAME].parts['Part-1'].Mirror(keepOriginal=ON, mirrorPlane=
mdb.models[MODELNAME].parts['Part-1'].faces[5])
mdb.models[MODELNAME].ConstrainedSketch(gridSpacing=0.07,
name='__profile__',
    sheetSize=3.0, transform=
    mdb.models[MODELNAME].parts['Part-1'].MakeSketchTransform(
    sketchPlane=mdb.models[MODELNAME].parts['Part-1'].faces[1],
    sketchPlaneSide=SIDE1,
    sketchUpEdge=mdb.models[MODELNAME].parts['Part-1'].edges[7],
    sketchOrientation=RIGHT, origin=(-0.5, 0.0, 0.0))
mdb.models[MODELNAME].parts['Part-1'].projectReferencesOntoSketch(filter=
COPLANAR_EDGES,
sketch=mdb.models[MODELNAME].sketches['__profile__'])
mdb.models[MODELNAME].sketches['__profile__'].CircleByCenterPerimeter(cente
r=(
    0.0, 0.0), point1=(R, 0.0))
mdb.models[MODELNAME].sketches['__profile__'].rectangle(point1=(-R/2, -
R/2),
    point2=(R/2, R/2))
mdb.models[MODELNAME].parts['Part-1'].PartitionCellBySketch(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz)
    , sketch=mdb.models[MODELNAME].sketches['__profile__'],
sketchPlane=
    mdb.models[MODELNAME].parts['Part-1'].faces[1], sketchUpEdge=
    mdb.models[MODELNAME].parts['Part-1'].edges[7])
del mdb.models[MODELNAME].sketches['__profile__']
mdb.models[MODELNAME].parts['Part-1'].PartitionCellByExtrudeEdge(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz)
    , edges=(mdb.models[MODELNAME].parts['Part-1'].edges[0], ), line=
    mdb.models[MODELNAME].parts['Part-1'].edges[11], sense=FORWARD)
mdb.models[MODELNAME].parts['Part-1'].PartitionCellByExtrudeEdge(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz)
    , edges=(mdb.models[MODELNAME].parts['Part-1'].edges[8],
    mdb.models[MODELNAME].parts['Part-1'].edges[9],
    mdb.models[MODELNAME].parts['Part-1'].edges[10],
    mdb.models[MODELNAME].parts['Part-1'].edges[11]), line=
    mdb.models[MODELNAME].parts['Part-1'].edges[12], sense=FORWARD)
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=mdb.models[MODELNAME].parts['Part
-1'].cells[0:2],
    point1=mdb.models[MODELNAME].parts['Part-1'].vertices[11], point2=
    mdb.models[MODELNAME].parts['Part-1'].vertices[14], point3=
    mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
    mdb.models[MODELNAME].parts['Part-1'].edges[22], MIDDLE))
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=mdb.models[MODELNAME].parts['Part
-1'].cells[0:4],
    point1=mdb.models[MODELNAME].parts['Part-1'].vertices[16], point2=
    mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(

```

```

        mdb.models[MODELNAME].parts['Part-1'].edges[39], MIDDLE), point3=
        mdb.models[MODELNAME].parts['Part-1'].vertices[23])
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=
        mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz),
        point1=mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
        mdb.models[MODELNAME].parts['Part-1'].edges[16], MIDDLE), point2=
        mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
        mdb.models[MODELNAME].parts['Part-1'].edges[37], MIDDLE), point3=
        mdb.models[MODELNAME].parts['Part-1'].InterestingPoint(
        mdb.models[MODELNAME].parts['Part-1'].edges[50], MIDDLE))
mdb.models[MODELNAME].parts['Part-
1'].PartitionCellByPlaneThreePoints(cells=
        mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz),
        point1=mdb.models[MODELNAME].parts['Part-1'].vertices[41],
        point2=mdb.models[MODELNAME].parts['Part-1'].vertices[40],
        point3=mdb.models[MODELNAME].parts['Part-1'].vertices[35])

mdb.models[MODELNAME].ConstrainedSketch(gridSpacing=0.03,
name='__profile__',
        sheetSize=1.41, transform=
        mdb.models[MODELNAME].parts['Part-1'].MakeSketchTransform(
        sketchPlane=mdb.models[MODELNAME].parts['Part-1'].faces[65],
        sketchPlaneSide=SIDE1,
        sketchUpEdge=mdb.models[MODELNAME].parts['Part-1'].edges[112],
        sketchOrientation=RIGHT, origin=(-0.5, 0, 0))
mdb.models[MODELNAME].parts['Part-1'].projectReferencesOntoSketch(filter=
        COPLANAR_EDGES, sketch=mdb.models[MODELNAME].sketches['__profile__'])
mdb.models[MODELNAME].sketches['__profile__'].CircleByCenterPerimeter(cente
r=(
        0.0,0.0), point1=(R+tc,0.0))
mdb.models[MODELNAME].sketches['__profile__'].CoincidentConstraint(addUndoS
tate=
        False,
entity1=mdb.models[MODELNAME].sketches['__profile__'].vertices[25],
        entity2=mdb.models[MODELNAME].sketches['__profile__'].geometry[28])
mdb.models[MODELNAME].parts['Part-1'].PartitionCellBySketch(cells=
        mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-Ly,-
Lz,Lx,Ly,Lz),
        sketch=mdb.models[MODELNAME].sketches['__profile__'], sketchPlane=
        mdb.models[MODELNAME].parts['Part-1'].faces[65], sketchUpEdge=
        mdb.models[MODELNAME].parts['Part-1'].edges[112])
del mdb.models[MODELNAME].sketches['__profile__']
mdb.models[MODELNAME].parts['Part-1'].PartitionCellByExtrudeEdge(cells=
        mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-Ly,-
Lz,Lx,Ly,Lz),
        edges=(mdb.models[MODELNAME].parts['Part-1'].edges[0],
        mdb.models[MODELNAME].parts['Part-1'].edges[4],
        mdb.models[MODELNAME].parts['Part-1'].edges[8],
        mdb.models[MODELNAME].parts['Part-1'].edges[11],
        mdb.models[MODELNAME].parts['Part-1'].edges[14],
        mdb.models[MODELNAME].parts['Part-1'].edges[17],
        mdb.models[MODELNAME].parts['Part-1'].edges[20],
        mdb.models[MODELNAME].parts['Part-1'].edges[23]), line=
        mdb.models[MODELNAME].parts['Part-1'].edges[126], sense=FORWARD)
mdb.models[MODELNAME].Material(name='Epoxy')
mdb.models[MODELNAME].materials['Epoxy'].Creep(dependencies=0, law=STRAIN,

```

```

        table=((1.10E-06, 1.0, -0.8218, 22.0), ), temperatureDependency=ON)
mdb.models[MODELNAME].materials['Epoxy'].setValues(materialIdentifier='')
mdb.models[MODELNAME].materials['Epoxy'].setValues(description='')
mdb.models[MODELNAME].materials['Epoxy'].Elastic(dependencies=0, moduli=
    LONG_TERM, noCompression=OFF, noTension=OFF, table=((4510.0, 0.311), ),
    temperatureDependency=OFF, type=ISOTROPIC)
mdb.models[MODELNAME].Material(name='Carbon Fiber')
mdb.models[MODELNAME].materials['Carbon
Fiber'].setValues(materialIdentifier='')
mdb.models[MODELNAME].materials['Carbon Fiber'].setValues(description='')
mdb.models[MODELNAME].materials['Carbon Fiber'].Elastic(dependencies=0,
moduli=
    LONG_TERM, noCompression=OFF, noTension=OFF, table=((202820.0, 25300.0,
    25300.0, 0.443, 0.443, 0.0, 44120.0, 44120.0, 44120.0), ),
    temperatureDependency=OFF, type=ENGINEERING_CONSTANTS)
mdb.models[MODELNAME].Material(name='INTERFASE')
mdb.models[MODELNAME].materials['INTERFASE'].Creep(dependencies=0,
law=STRAIN,
    table=((1.10E-06, 1.0, -0.8218, 22.0), ), temperatureDependency=ON)
mdb.models[MODELNAME].materials['INTERFASE'].setValues(materialIdentifier='
')
mdb.models[MODELNAME].materials['INTERFASE'].setValues(description='')
mdb.models[MODELNAME].materials['INTERFASE'].Elastic(dependencies=0,
moduli=
    LONG_TERM, noCompression=OFF, noTension=OFF, table=((4510.0, 0.311), ),
    temperatureDependency=OFF, type=ISOTROPIC)

mdb.models[MODELNAME].HomogeneousSolidSection(material='Carbon Fiber',
name=
    'FIBER', thickness=None)
mdb.models[MODELNAME].HomogeneousSolidSection(material='Epoxy',
name='MATRIX',
    thickness=None)
FIBER=mdb.models[MODELNAME].parts['Part-1'].Set(cells=
    mdb.models[MODELNAME].parts['Part-
1'].cells.getByBoundingCylinder((-Lx,0.0,0.0),(Lx,0.0,0.0),R+tol),
name='FIBER')
COMPOSITE=mdb.models[MODELNAME].parts['Part-1'].Set(cells=
    mdb.models[MODELNAME].parts['Part-1'].cells.getByBoundingBox(-Lx,-
Ly,-Lz,Lx,Ly,Lz), name='COMPOSITE')
MATRIX=mdb.models[MODELNAME].parts['Part-
1'].SetByBoolean(name='MATRIX',sets=(COMPOSITE,FIBER),operation=DIFFERENCE)

mdb.models[MODELNAME].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models[MODELNAME].parts['Part-1'].sets['FIBER'],
sectionName='FIBER',
    thicknessAssignment=FROM_SECTION)
mdb.models['M-1'].HomogeneousSolidSection(material='INTERFASE', name=
    'INTERFASE', thickness=None)
mdb.models['M-1'].parts['Part-1'].Set(cells=
    mdb.models['M-1'].parts['Part-1'].cells.getSequenceFromMask(('[#7f80
]', ), ),
    ), name='Set-5')
mdb.models['M-1'].parts['Part-1'].SectionAssignment(offset=0.0,
offsetField='',
    offsetType=MIDDLE_SURFACE, region=
    mdb.models['M-1'].parts['Part-1'].sets['Set-5'],
sectionName='INTERFASE',

```

```

thicknessAssignment=FROM_SECTION)

mdb.models[MODELNAME].parts['Part-1'].Set(cells=
    mdb.models[MODELNAME].parts['Part-
1'].cells.getSequenceFromMask(['#807f ', ],
    ), name='Set-6')
mdb.models[MODELNAME].parts['Part-1'].SectionAssignment(offset=0.0,
offsetField='',
    offsetType=MIDDLE_SURFACE, region=
    mdb.models[MODELNAME].parts['Part-1'].sets['Set-6'],
sectionName='MATRIX',
    thicknessAssignment=FROM_SECTION)

mdb.models[MODELNAME].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models[MODELNAME].rootAssembly.Instance(dependent=ON, name='Part-1-1',
    part=mdb.models[MODELNAME].parts['Part-1'])
mdb.models[MODELNAME].rootAssembly.translate(instanceList=('Part-1-1', ),
    vector=(0.0, (1-ny)*Ly/2, (1-nz)*Lz/2))
if (ny>1 or nz>1):
    mdb.models[MODELNAME].rootAssembly.LinearInstancePattern(direction1
=(0.0, 1.0,
        0.0), direction2=(0.0, 0.0, 1.0), instanceList=('Part-1-1',
), number1=ny,
        number2=nz, spacing1=1.0, spacing2=1.0)
    my_list = []
    RA=mdb.models[MODELNAME].rootAssembly
    for yi in range(ny):
        for zi in range(nz):
            if (yi==0 and zi==0):
                name = RA.instances[str('Part-1-1')]
            else:
                name = RA.instances[str('Part-1-1-lin-
')+str(yi+1)+str('-')+str(zi+1)]
                my_list.append(name)
    my_list = tuple(my_list)
    mdb.models[MODELNAME].rootAssembly.InstanceFromBooleanMerge(domain=
BOTH,
        instances=my_list,keepIntersections=ON,
mergeNodes=BOUNDARY_ONLY, name='Part-2',
        originalInstances=DELETE)
    mdb.models[MODELNAME].parts['Part-
2'].seedEdgeBySize(constraint=FINER,
        deviationFactor=ESize, edges=
        mdb.models[MODELNAME].parts['Part-
2'].edges.getByBoundingBox(-100, -100, -100, 100,100, 100), size=ESize)
    mdb.models[MODELNAME].parts['Part-2'].generateMesh()
    PName=str('Part-2')
else:
    mdb.models[MODELNAME].parts['Part-
1'].seedEdgeBySize(constraint=FINER,
        deviationFactor=ESize, edges=
        mdb.models[MODELNAME].parts['Part-
1'].edges.getByBoundingBox(-100, -100, -100, 100,100, 100), size=ESize)
    mdb.models[MODELNAME].parts['Part-1'].generateMesh()
    PName=str('Part-1')
mdb.models[MODELNAME].rootAssembly.regenerate()

```

```

mdb.models[MODELNAME].ViscoStep(cetol=0.001, initialInc=72.0,
maxInc=7200.0,
    maxNumInc=1000000, minInc=0.072, name='Step-1', nlgeom=ON, previous=
    'Initial', timePeriod=7200.0)
mdb.models[MODELNAME].TimePoint(name='TimePoints-1', points=((0.0, 100.0,
20.0), (
    100.0, 1000.0, 100.0), (1000.0, 10000.0, 500.0)))
mdb.models[MODELNAME].fieldOutputRequests['F-Output-
1'].setValues(timePoint=
    'TimePoints-1', variables=('S', 'E', 'PE', 'PEEQ', 'PEMAG', 'EE', 'CE',
    'CEP', 'CEEQ', 'LE', 'U', 'RF', 'CF', 'CSTRESS', 'CDISP', 'ENER',
    'ELEN',
    'ELEDEN', 'EVOL'))

mdb.models[MODELNAME].parts[PName].MaterialOrientation(
    additionalRotationType=ROTATION_NONE, axis=AXIS_1, fieldName='',
localCsys=
    None, orientationType=GLOBAL, region=
    mdb.models[MODELNAME].parts[PName].sets['FIBER'], stackDirection=
    STACK_3)
mdb.models[MODELNAME].parts[PName].MaterialOrientation(
    additionalRotationType=ROTATION_NONE, axis=AXIS_1, fieldName='',
localCsys=
    None, orientationType=GLOBAL, region=
    mdb.models[MODELNAME].parts[PName].sets['MATRIX'], stackDirection=
    STACK_3)

#*****
#*****
# ***** Create Reference Point Set
#*****
#
#*****
#*****
RP_E11=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E22=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E33=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E12=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E13=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))
RP_E23=mdb.models[MODELNAME].rootAssembly.ReferencePoint(point=(0.5, 0.0,
0.0))

mdb.models[MODELNAME].rootAssembly.Set(name='RP_E11', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E11.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E22', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E22.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E33', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E33.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E12', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E12.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E13', referencePoints=(

```

```

        mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E13.id], ))
mdb.models[MODELNAME].rootAssembly.Set(name='RP_E23', referencePoints=(
    mdb.models[MODELNAME].rootAssembly.referencePoints[RP_E23.id], ))
#*****
#*****
# ***** Create Boundry Condition Node Set
#*****
#
#*****
#*****
# ***** Ceaate Set for node on the surface but not on the edges and
corner *****
XN=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,-c+tol,-a+tol,b+tol,c+tol)
XP=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a+tol,-b+tol,-c+tol,a+tol,b+tol,c+tol)
YN=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,-c+tol,a+tol,-b+tol,c+tol)
YP=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,b+tol,-c+tol,a+tol,b+tol,c+tol)
ZN=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,-c+tol,a+tol,b+tol,-c+tol)
ZP=mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b+tol,c+tol,a+tol,b+tol,c+tol)
mdb.models[MODELNAME].rootAssembly.Set(name='XN', nodes=XN)
mdb.models[MODELNAME].rootAssembly.Set(name='XP', nodes=XP)
mdb.models[MODELNAME].rootAssembly.Set(name='YN', nodes=YN)
mdb.models[MODELNAME].rootAssembly.Set(name='YP', nodes=YP)
mdb.models[MODELNAME].rootAssembly.Set(name='ZN', nodes=ZN)
mdb.models[MODELNAME].rootAssembly.Set(name='ZP', nodes=ZP)
for IXN in range (len(XN)):
    for IXP in range (len(XP)):
        if (abs(XN[IXN].coordinates[1]-
XP[IXP].coordinates[1])<tol) and (abs(XN[IXN].coordinates[2]-
XP[IXP].coordinates[2])<tol):
            mdb.models[MODELNAME].rootAssembly.Set(name='XN-
'+str(IXN+1), nodes=XN[IXN:IXN+1])
            mdb.models[MODELNAME].rootAssembly.Set(name='XP-
'+str(IXN+1), nodes= XP[IXP:IXP+1])
for IYN in range (len(YN)):
    for IYP in range (len(YP)):
        if (abs(YN[IYN].coordinates[0]-
YP[IYP].coordinates[0])<tol) and (abs(YN[IYN].coordinates[2]-
YP[IYP].coordinates[2])<tol):
            mdb.models[MODELNAME].rootAssembly.Set(name='YN-
'+str(IYN+1), nodes=YN[IYN:IYN+1])
            mdb.models[MODELNAME].rootAssembly.Set(name='YP-
'+str(IYN+1), nodes= YP[IYP:IYP+1])
for IZN in range (len(ZN)):
    for IZP in range (len(ZP)):
        if (abs(ZN[IZN].coordinates[0]-
ZP[IZP].coordinates[0])<tol) and (abs(ZN[IZN].coordinates[1]-
ZP[IZP].coordinates[1])<tol):
            mdb.models[MODELNAME].rootAssembly.Set(name='ZN-
'+str(IZN+1), nodes=ZN[IZN:IZN+1])
            mdb.models[MODELNAME].rootAssembly.Set(name='ZP-
'+str(IZN+1), nodes= ZP[IZP:IZP+1])

for IX in range(len(XN)):

```

```

        mdb.models[MODELNAME].Equation(name='SCEXX-'+str(IX+1),
terms=((-1.0, 'XP-'+str(IX+1), 1),
        (1.0, 'XN-'+str(IX+1), 1), (1, 'RP_E11', 1)))
        mdb.models[MODELNAME].Equation(name='SCEXY-'+str(IX+1),
terms=((-1.0, 'XP-'+str(IX+1), 2),
        (1.0, 'XN-'+str(IX+1), 2), (1, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='SCEXZ-'+str(IX+1),
terms=((-1.0, 'XP-'+str(IX+1), 3),
        (1.0, 'XN-'+str(IX+1), 3), (1, 'RP_E13', 1)))
for IY in range(len(YN)):
        mdb.models[MODELNAME].Equation(name='SCEYX-'+str(IY+1),
terms=((-1.0, 'YP-'+str(IY+1), 1),
        (1.0, 'YN-'+str(IY+1), 1), (b/a, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='SCEYY-'+str(IY+1),
terms=((-1.0, 'YP-'+str(IY+1), 2),
        (1.0, 'YN-'+str(IY+1), 2), (1, 'RP_E22', 1)))
        mdb.models[MODELNAME].Equation(name='SCEYZ-'+str(IY+1),
terms=((-1.0, 'YP-'+str(IY+1), 3),
        (1.0, 'YN-'+str(IY+1), 3), (1, 'RP_E23', 1)))
for IZ in range(len(ZN)):
        mdb.models[MODELNAME].Equation(name='SCEZX-'+str(IZ+1),
terms=((-1.0, 'ZP-'+str(IZ+1), 1),
        (1.0, 'ZN-'+str(IZ+1), 1), (c/a, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='SCEZY-'+str(IZ+1),
terms=((-1.0, 'ZP-'+str(IZ+1), 2),
        (1.0, 'ZN-'+str(IZ+1), 2), (c/b, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='SCEZZ-'+str(IZ+1),
terms=((-1.0, 'ZP-'+str(IZ+1), 3),
        (1.0, 'ZN-'+str(IZ+1), 3), (1, 'RP_E33', 1)))
#
*****
*****
# ***** Ceaate Set for node  on the edges but not on
corner *****
XYNN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b-tol,-c+tol,-a+tol,-b+tol,c-tol)
XYNP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,b-tol,-c+tol,-a+tol,b+tol,c-tol)
XZNN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b+tol,-c-tol,-a+tol,b-tol,-c+tol)
XZNP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b+tol,c-tol,-a+tol,b-tol,c+tol)
XYPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b-tol,-c+tol,a+tol,-b+tol,c-tol)
XYPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,b-tol,-c+tol,a+tol,b+tol,c-tol)
XZPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b+tol,-c-tol,a+tol,b-tol,-c+tol)
XZPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b+tol,c-tol,a+tol,b-tol,c+tol)
YZNN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b-tol,-c-tol,a-tol,-b+tol,-c+tol)
YZNP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,-b-tol,c-tol,a-tol,-b+tol,c+tol)
YZPN= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,b-tol,-c-tol,a-tol,b+tol,-c+tol)
YZPP= mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a+tol,b-tol,c-tol,a-tol,b+tol,c+tol)
mdb.models[MODELNAME].rootAssembly.Set(name='XYNN', nodes=XYNN)

```

```

mdb.models[MODELNAME].rootAssembly.Set(name='XYPN', nodes=XYPN)
mdb.models[MODELNAME].rootAssembly.Set(name='XZNN', nodes=XZNN)
mdb.models[MODELNAME].rootAssembly.Set(name='XZNP', nodes=XZNP)
mdb.models[MODELNAME].rootAssembly.Set(name='XYPN', nodes=XYPN)
mdb.models[MODELNAME].rootAssembly.Set(name='XYPP', nodes=XYPP)
mdb.models[MODELNAME].rootAssembly.Set(name='XZPN', nodes=XZPN)
mdb.models[MODELNAME].rootAssembly.Set(name='XZPP', nodes=XZPP)
mdb.models[MODELNAME].rootAssembly.Set(name='YZNN', nodes=YZNN)
mdb.models[MODELNAME].rootAssembly.Set(name='YZNP', nodes=YZNP)
mdb.models[MODELNAME].rootAssembly.Set(name='YZPN', nodes=YZPN)
mdb.models[MODELNAME].rootAssembly.Set(name='YZPP', nodes=YZPP)

for IXYNN in range (len(XYNN)):
    for IXYPN in range (len(XYPN)):
        if (abs(XYNN[IXYNN].coordinates[2]-
XYNP[IXYPN].coordinates[2])<tol):
            for IXYPN in range (len(XYPN)):
                if (abs(XYNN[IXYNN].coordinates[2]-
XYPN[IXYPN].coordinates[2])<tol):
                    for IXYPP in range (len(XYPP)):
                        if
(abs(XYNN[IXYNN].coordinates[2]-XYPP[IXYPP].coordinates[2])<tol):

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYNN-'+str(IXYNN+1),
nodes=XYNN[IXYNN:IXYNN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYPN-'+str(IXYNN+1),
nodes=XYPN[IXYPN:IXYPN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYPN-'+str(IXYNN+1),
nodes=XYPN[IXYPN:IXYPN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XYPP-'+str(IXYNN+1),
nodes=XYPP[IXYPP:IXYPP+1])
for IXZNN in range (len(XZNN)):
    for IXZNP in range (len(XZNP)):
        if (abs(XZNN[IXZNN].coordinates[1]-
XZNP[IXZNP].coordinates[1])<tol):
            for IXZPN in range (len(XZPN)):
                if (abs(XZNN[IXZNN].coordinates[1]-
XZPN[IXZPN].coordinates[1])<tol):
                    for IXZPP in range (len(XZPP)):
                        if
(abs(XZNN[IXZNN].coordinates[1]-XZPP[IXZPP].coordinates[1])<tol):

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZNN-'+str(IXZNN+1),
nodes=XZNN[IXZNN:IXZNN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZNP-'+str(IXZNN+1),
nodes=XZNP[IXZNP:IXZNP+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZPN-'+str(IXZNN+1),
nodes=XZPN[IXZPN:IXZPN+1])

                            mdb.models[MODELNAME].rootAssembly.Set(name='XZPP-'+str(IXZNN+1),
nodes=XZPP[IXZPP:IXZPP+1])
for IYZNN in range (len(YZNN)):
    for IYZNP in range (len(YZNP)):

```

```

        if (abs(YZNN[IYZNN].coordinates[0]-
YZNP[IYZNP].coordinates[0])<tol):
            for IYZPN in range (len(YZPN)):
                if (abs(YZNN[IYZNN].coordinates[0]-
YZPN[IYZPN].coordinates[0])<tol):
                    for IYZPP in range (len(YZPP)):
                        if
(abs(YZNN[IYZNN].coordinates[0]-YZPP[IYZPP].coordinates[0])<tol):

                            mdb.models[MODELNAME].rootAssembly.Set (name='YZNN-'+str(IYZNN+1),
nodes=YZNN[IYZNN:IYZNN+1])

                            mdb.models[MODELNAME].rootAssembly.Set (name='YZNP-'+str(IYZNN+1),
nodes=YZNP[IYZNP:IYZNP+1])

                            mdb.models[MODELNAME].rootAssembly.Set (name='YZPN-'+str(IYZNN+1),
nodes=YZPN[IYZPN:IYZPN+1])

                            mdb.models[MODELNAME].rootAssembly.Set (name='YZPP-'+str(IYZNN+1),
nodes=YZPP[IYZPP:IYZPP+1])

for IXY in range(len(XYNN)):
    mdb.models[MODELNAME].Equation(name='ECXY_X1-'+str(IXY+1),
terms=((-1.0, 'XYPP-'+str(IXY+1),1),
        (1.0, 'XYPN-'+str(IXY+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_X2-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),1),
        (1.0, 'XYNN-'+str(IXY+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_X3-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),1),
        (1.0, 'XYNN-'+str(IXY+1), 1),(b/a, 'RP_E12', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Y1-'+str(IXY+1),
terms=((-1.0, 'XYPP-'+str(IXY+1),2),
        (1.0, 'XYPN-'+str(IXY+1), 2),(1, 'RP_E12', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Y2-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),2),
        (1.0, 'XYNN-'+str(IXY+1), 2),(1, 'RP_E12', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Y3-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),2),
        (1.0, 'XYNN-'+str(IXY+1), 2),(1, 'RP_E22', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Z1-'+str(IXY+1),
terms=((-1.0, 'XYPP-'+str(IXY+1),3),
        (1.0, 'XYPN-'+str(IXY+1), 3),(1, 'RP_E13', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Z2-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),3),
        (1.0, 'XYNN-'+str(IXY+1), 3),(1, 'RP_E13', 1)))
    mdb.models[MODELNAME].Equation(name='ECXY_Z3-'+str(IXY+1),
terms=((-1.0, 'XYPN-'+str(IXY+1),3),
        (1.0, 'XYNN-'+str(IXY+1), 3),(1, 'RP_E23', 1)))
for IXZ in range(len(XZNN)):
    mdb.models[MODELNAME].Equation(name='ECXZ_X1-'+str(IXZ+1),
terms=((-1.0, 'XZPP-'+str(IXZ+1),1),
        (1.0, 'XZNP-'+str(IXZ+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXZ_X2-'+str(IXZ+1),
terms=((-1.0, 'XZPN-'+str(IXZ+1),1),
        (1.0, 'XZNN-'+str(IXZ+1), 1),(1, 'RP_E11', 1)))
    mdb.models[MODELNAME].Equation(name='ECXZ_X3-'+str(IXZ+1),
terms=((-1.0, 'XZNP-'+str(IXZ+1),1),
        (1.0, 'XZNN-'+str(IXZ+1), 1),(c/a, 'RP_E13', 1)))

```

```

        mdb.models[MODELNAME].Equation(name='ECXZ_Y1-'+str(IXZ+1),
terms=((-1.0, 'XZPP-'+str(IXZ+1), 2),
        (1.0, 'XZNP-'+str(IXZ+1), 2), (1, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Y2-'+str(IXZ+1),
terms=((-1.0, 'XZPN-'+str(IXZ+1), 2),
        (1.0, 'XZNN-'+str(IXZ+1), 2), (1, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Y3-'+str(IXZ+1),
terms=((-1.0, 'XZNP-'+str(IXZ+1), 2),
        (1.0, 'XZNN-'+str(IXZ+1), 2), (c/b, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Z1-'+str(IXZ+1),
terms=((-1.0, 'XZPP-'+str(IXZ+1), 3),
        (1.0, 'XZNP-'+str(IXZ+1), 3), (1, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Z2-'+str(IXZ+1),
terms=((-1.0, 'XZPN-'+str(IXZ+1), 3),
        (1.0, 'XZNN-'+str(IXZ+1), 3), (1, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='ECXZ_Z3-'+str(IXZ+1),
terms=((-1.0, 'XZNP-'+str(IXZ+1), 3),
        (1.0, 'XZNN-'+str(IXZ+1), 3), (1, 'RP_E33', 1)))
for IYZ in range(len(YZNN)):
        mdb.models[MODELNAME].Equation(name='ECYZ_X1-'+str(IYZ+1),
terms=((-1.0, 'YZPP-'+str(IYZ+1), 1),
        (1.0, 'YZNP-'+str(IYZ+1), 1), (b/a, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_X2-'+str(IYZ+1),
terms=((-1.0, 'YZPN-'+str(IYZ+1), 1),
        (1.0, 'YZNN-'+str(IYZ+1), 1), (b/a, 'RP_E12', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_X3-'+str(IYZ+1),
terms=((-1.0, 'YZNP-'+str(IYZ+1), 1),
        (1.0, 'YZNN-'+str(IYZ+1), 1), (c/a, 'RP_E13', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Y1-'+str(IYZ+1),
terms=((-1.0, 'YZPP-'+str(IYZ+1), 2),
        (1.0, 'YZNP-'+str(IYZ+1), 2), (1, 'RP_E22', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Y2-'+str(IYZ+1),
terms=((-1.0, 'YZPN-'+str(IYZ+1), 2),
        (1.0, 'YZNN-'+str(IYZ+1), 2), (1, 'RP_E22', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Y3-'+str(IYZ+1),
terms=((-1.0, 'YZNP-'+str(IYZ+1), 2),
        (1.0, 'YZNN-'+str(IYZ+1), 2), (c/b, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Z1-'+str(IYZ+1),
terms=((-1.0, 'YZPP-'+str(IYZ+1), 3),
        (1.0, 'YZNP-'+str(IYZ+1), 3), (1, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Z2-'+str(IYZ+1),
terms=((-1.0, 'YZPN-'+str(IYZ+1), 3),
        (1.0, 'YZNN-'+str(IYZ+1), 3), (1, 'RP_E23', 1)))
        mdb.models[MODELNAME].Equation(name='ECYZ_Z3-'+str(IYZ+1),
terms=((-1.0, 'YZNP-'+str(IYZ+1), 3),
        (1.0, 'YZNN-'+str(IYZ+1), 3), (1, 'RP_E33', 1)))
#
*****
*****
# ***** Ceaate Set for node on the corner
*****
mdb.models[MODELNAME].rootAssembly.Set(name='NNN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')] .nodes.getByBoundingBox(-a-tol, -b-tol, -c-tol, -a+tol, -b+tol, -c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='NPN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')] .nodes.getByBoundingBox(-a-tol, b-tol, -c-tol, -a+tol, b+tol, -c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='NPP', nodes=

```

```

        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,b-tol,c-tol,-a+tol,b+tol,c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='NNP', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(-a-tol,-b-tol,c-tol,-a+tol,-b+tol,c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PNN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b-tol,-c-tol,a+tol,-b+tol,-c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PPN', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,b-tol,-c-tol,a+tol,b+tol,-c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PPP', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,b-tol,c-tol,a+tol,b+tol,c+tol))
mdb.models[MODELNAME].rootAssembly.Set(name='PNP', nodes=
        mdb.models[MODELNAME].rootAssembly.instances[PName+str('-
1')].nodes.getByBoundingBox(a-tol,-b-tol,c-tol,a+tol,-b+tol,c+tol))

mdb.models[MODELNAME].Equation(name='CX1', terms=((-1.0,'PPP',1),
(1.0, 'NPP', 1),(1, 'RP_E11', 1)))
mdb.models[MODELNAME].Equation(name='CX2', terms=((-1.0,'PNP',1),
(1.0, 'NNP', 1),(1, 'RP_E11', 1)))
mdb.models[MODELNAME].Equation(name='CX3', terms=((-1.0,'PNN',1),
(1.0, 'NNN', 1),(1, 'RP_E11', 1)))
mdb.models[MODELNAME].Equation(name='CX4', terms=((-1.0,'PPN',1),
(1.0, 'PNN', 1),(b/a, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CX5', terms=((-1.0,'NPN',1),
(1.0, 'NNN', 1),(b/a, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CX6', terms=((-1.0,'NPP',1),
(1.0, 'NPN', 1),(c/a, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CX7', terms=((-1.0,'NNP',1),
(1.0, 'NNN', 1),(c/a, 'RP_E13', 1)))

mdb.models[MODELNAME].Equation(name='CY1', terms=((-1.0,'PPP',2),
(1.0, 'NPP', 2),(1, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CY2', terms=((-1.0,'PNP',2),
(1.0, 'NNP', 2),(1, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CY3', terms=((-1.0,'PNN',2),
(1.0, 'NNN', 2),(1, 'RP_E12', 1)))
mdb.models[MODELNAME].Equation(name='CY4', terms=((-1.0,'PPN',2),
(1.0, 'PNN', 2),(1, 'RP_E22', 1)))
mdb.models[MODELNAME].Equation(name='CY5', terms=((-1.0,'NPN',2),
(1.0, 'NNN', 2),(1, 'RP_E22', 1)))
mdb.models[MODELNAME].Equation(name='CY6', terms=((-1.0,'NPP',2),
(1.0, 'NPN', 2),(c/b, 'RP_E23', 1)))
mdb.models[MODELNAME].Equation(name='CY7', terms=((-1.0,'NNP',2),
(1.0, 'NNN', 2),(c/b, 'RP_E23', 1)))

mdb.models[MODELNAME].Equation(name='CZ1', terms=((-1.0,'PPP',3),
(1.0, 'NPP', 3),(1, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CZ2', terms=((-1.0,'PNP',3),
(1.0, 'NNP', 3),(1, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CZ3', terms=((-1.0,'PNN',3),
(1.0, 'NNN', 3),(1, 'RP_E13', 1)))
mdb.models[MODELNAME].Equation(name='CZ4', terms=((-1.0,'PPN',3),
(1.0, 'PNN', 3),(1, 'RP_E23', 1)))
mdb.models[MODELNAME].Equation(name='CZ5', terms=((-1.0,'NPN',3),
(1.0, 'NNN', 3),(1, 'RP_E23', 1)))

```

```

mdb.models[MODELNAME].Equation(name='CZ6', terms=((-1.0, 'NPP', 3),
(1.0, 'NPN', 3), (1, 'RP_E33', 1)))
mdb.models[MODELNAME].Equation(name='CZ7', terms=((-1.0, 'NNP', 3),
(1.0, 'NNN', 3), (1, 'RP_E33', 1)))
#####

if IBC==1:
    BC_SET='RP_E11'
    CF=1.0*b*c*4*1.0

elif IBC==2:
    BC_SET='RP_E22'
    CF=1.0*a*c*4*1.0

elif IBC==3:
    BC_SET='RP_E33'
    CF=1.0*b*a*4*1.0

elif IBC==4:#Sxy
    BC_SET='RP_E12'
    CF=2.0*b*c*4*1.0

elif IBC==5:#Sxz
    BC_SET='RP_E13'
    CF=2.0*b*c*4*1.0

else:#Syz
    BC_SET='RP_E23'
    CF=2.0*a*c*4*1.0

#
*****
*****
#***** Boundary Conditions
*****
#
*****
*****
mdb.models[MODELNAME].TabularAmplitude(name='Amp-1', timeSpan=STEP,
smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1.0, 1.0), (7200.0, 1.0)))
mdb.models[MODELNAME].ConcentratedForce(name='Load-1', cfl=CF,
createStepName='Step-1', amplitude='Amp-1',
distributionType=UNIFORM, field='', localCsys=None,
region=mdb.models[MODELNAME].rootAssembly.sets[BC_SET])
# mdb.models[MODELNAME].ConcentratedForce(cfl=CF, createStepName='Step-1',
# distributionType=UNIFORM, field='', localCsys=None, name='Load-
1', region=
# mdb.models[MODELNAME].rootAssembly.sets[BC_SET])
#
*****
*****
#***** Job Genration
*****
#
*****
*****

```

```

Job_Name=MODELNAME+'-'+str(IBC)
mdb.models[MODELNAME].historyOutputRequests['H-Output-1'].setValues(timePoint='TimePoints-1')
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF, explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF, memory=50, memoryUnits=PERCENTAGE, model=MODELNAME, modelPrint=OFF, multiprocessingMode=DEFAULT, name=Job_Name, nodalOutputPrecision=SINGLE, numCpus=2, numDomains=2, queue=None, scratch='', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs[Job_Name].submit(consistencyChecking=OFF)
mdb.jobs[Job_Name].waitForCompletion()
#####

```

```

odb = session.openOdb(Job_Name+'.odb')

Ese = session.XYDataFromHistory(name='StrainEnergy',
odb=odb,outputVariableName='Strain energy: ALLSE for Whole Model',
steps=('Step-1',), )
#StrainEnergy=Ese[len(xy_result)-1][1]
# SE = session.Curve(xyData=Ese).data
# outputF1 = open( Job_Name + '_Ese.csv', 'wb')
# writer = csv.writer(outputF1)
# writer.writerows(SE)
# outputF1.close()
Cde = session.XYDataFromHistory(name='ALLCD Whole Model-1',
odb=odb,outputVariableName='Creep dissipation energy: ALLCD for Whole Model',steps=('Step-1', ),)
DAT=(Ese+Cde)*2.0 # 2U=Sigma^2/E
with open(Job_Name + '_Ese.plt', 'wb') as f:
    np.savetxt(f,Ese,delimiter=' ')
with open(Job_Name + '_Cde.plt', 'wb') as f:
    np.savetxt(f,Cde,delimiter=' ')
with open(Job_Name + '_Te.plt', 'wb') as f:
    np.savetxt(f,DAT,delimiter=' ')

```

پیوست د) حالت تولید رندم

```
# Script for Random Fiber Placement
# Christopher Cater, Michigan State University 2012
# Date of last modification: 3/1/2013

# Code Notes
# The code is split into two parts.
# 1) Determine fiber locations, allowing some interpenetration of fibers
# 2) Perturb the fibers until there is no interpenetration
# Import necessary modules
import csv #For reading/writing CSV files
from math import pi
from math import pow
from math import cos
from math import sin
from numpy import zeros
from numpy import array
import random

##### CODE INPUT
#####
# numFibers = 8                # Total number of fibers
# volFraction = 0.45          # Volume fraction target with set
# number of fibers
# Lx=1.0                      #Length Per X Edge
# Ly=1.0                      #Length Per Y Edge
# Lz=1.0                      #Length Per Z Edge
# penTol = 0.1                #Initial tolerance for penetration as
# a fraction of fiber radius
# penTo = 0.1

#Placeholder for current x and y coordinates
# xCord = 0.0
# yCord = 0.0

#Fiber radius (based on square array and desired volume fraction)
#fibRadius = round(pow(((volFraction*Lx*Ly)/(pi*numFibers)),0.5),2)

##### DEFINE FUNCTIONS
#####

#Define Functions
def xCoord(node):
    return centerCoords[node-1][0]
def yCoord(node):
    return centerCoords[node-1][1]
def distance(node1,node2):
    tmp = pow((xCoord(node1)-xCoord(node2))**2.0+(yCoord(node1)-
yCoord(node2))**2.0,0.5)
```

```

        return tmp
def distanceCoords(x1,y1,x2,y2):
    tmp = pow((x1-x2)**2.0+(y1-y2)**2.0,0.5)
    return tmp
def unitVector(node1,node2):
    mag = distance(node1,node2)
    z = [(xCoord(node1)-xCoord(node2))/mag, (yCoord(node1)-
yCoord(node2))/mag]
    return z
def multVector(vector,scalar):
    for i in range(0,len(vector)):
        vector[i] = scalar*vector[i]
    return vector

def inList(a,b):
    inListFlag = 0
    for row in b:
        if a==row:
            inListFlag=1
    if inListFlag==1:
        return True
    else:
        return False

def deleteRow(a,b):
    tempHolder=[]
    for i in range(0,len(a)):
        if i!=b:
            tempHolder.append(a[i])
    return tempHolder
#Check if a fiber exists outside of the RVE
# Tests all edges in one test defined below:
def outsideRVE(fiber):
    if (xCoord(fiber)>(Lx/2.0-fibRadius) or xCoord(fiber)<=
Lx/2.0+fibRadius
        or yCoord(fiber)>(Ly/2.0-fibRadius) or yCoord(fiber)<=
Ly/2.0+fibRadius):
        return True
    else:
        return False
def checkNonZero(array):
    rows = len(array)
    columns = len(array[0])
    testZero = 0
    for i in range(0,rows):
        for j in range(0,columns):
            if array[i][j]!=0:
                testZero=1
    if testZero==1:
        return True
    elif testZero==0:
        return False
def checkNonZeroVector(vector):
    columns = len(vector)
    testZero = 0
    for i in range(0,columns):
        if vector[i]!=0:
            testZero=1

```

```

        if testZero==1:
            return True
        elif testZero==0:
            return False
fiberNumber = 1
testNumber = 0
while fiberNumber < (numFibers)+1:
    testNumber = testNumber+1
    ##### DETERMINE FIBER PLACEMENTS (INITIAL)
    #####
    #Begin iterations
    #Set iteration parameters
    instVector = zeros([numFibers,4])
    fiberLink=[1]
    fiberNumber = 2
    iteration = 1
    numPenFibers = 0 #Placeholder for intial number of instances of
penetrating fibers
    #First Fiber (completely within RVE)
    centerCoords = [[random.uniform(-
Lx/2.0+fibRadius*(1.0+penTo),Lx/2.0-fibRadius*(1.0+penTo)),
                    random.uniform(-
Ly/2.0+fibRadius*(1.0+penTo),Ly/2.0-fibRadius*(1.0+penTo))]]
    #Loop based on fiber number
    while fiberNumber < (numFibers)+1:
        #Generate random fiber
        xcord = random.uniform(-
Lx/2.0+fibRadius*(1.0+penTo),Lx/2.0-fibRadius*(1.0+penTo))
        ycord = random.uniform(-
Ly/2.0+fibRadius*(1.0+penTo),Ly/2.0-fibRadius*(1.0+penTo))
        #Test for clearance within tolerance other fibers
        flag = 0
        #Debug output

        for i in range(1,len(centerCoords)+1):
            test =
distanceCoords(xcord,ycord,xCoord(i),yCoord(i))

            if test > ((1.0+penTol)*fibRadius*2.0):
                # if test < 2*fibRadius:
                # numPenFibers=numPenFibers+1
                flag = flag+0
            elif test<=((1.0+penTol)*fibRadius*2.0):
                flag = flag+1
    if flag==0:
        #Test for fiber on RVE edge
        toAdd = [xcord,ycord]
        centerCoords.append(toAdd)
        print 'Fiber Number: ', fiberNumber
        fiberNumber = fiberNumber+1
        print 'Number of iterations: ', iteration
        print 'Number of tests: ', testNumber
        iteration = 1
    else:
        iteration = iteration+1
    if iteration > 500:
        print 'Max iterations exceeded'
        break
if testNumber > 1000:

```

```

        print 'Max numbr test exceeded'
        break
# print fiberLink
# print centerCoords
# print instVector

oldarrayCenterCoords=array(centerCoords)

#Output array to CSV file
writeFile = open('randomFiber1.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(oldarrayCenterCoords)
writeFile.close()

```

Abstract

The use of composite materials in the design of main structures is increasing in various industries. In polymeric composite materials, due to the viscoelastic behavior of the matrix phase, this phase undergoes the phenomenon of creep and stress release even at low temperatures. For this reason, considering the creep phenomena in the design of composite structures is vital.

In this research, first, different types of scales used for the analysis of composites are introduced. Then, different creep models are presented. Then the micromechanical model is used to investigate the creep phenomenon. Modeling has been done for three models of single fiber and single fiber model with intermediate phase and fiber distribution model. In order to predict the properties of the composite in different volume fractions, modeling has been done with different volume fractions. All three models were tested under a periodic boundary conditions for 7200 seconds at a load of 1 MPa and their accuracy was verified by experimental results. Finally, the strain coefficient in different volume fractions was investigated. In the single-fiber mode, the strain decreases with increasing fiber volume fractions and because in the first direction the properties of the material are more dependent on the properties of the fibers, in this direction the changes in properties with temperature are negligible. In the single-fiber mode with intermediate phase, the failure parameter d is used to model the damage in the intermediate phase. The results show that with increasing the failure parameter, the effective modulus of the composite decreases and the effect of this parameter on the reduction behavior of the transverse and shear modulus is greater than the longitudinal modulus of the composite. In the case of random distribution of fibers, the obtained results show a good agreement between this case and the single-fiber mode.

Keywords: Composite, creep, micromechanics, volume fraction, power law, Representative Volumetric Element



Shahrood University of Technology

Faculty of Mechanical and Mechatronics Engineering

M.Sc. Thesis in Aerospace - Space Structures Engineering

Prediction of the creep behavior of unidirectional polymer composites by micromechanical model

By:

Sahar Ebrahimi

Supervisors:

Dr. Mehdi Ghannad

Dr. Hosein Toozandehjani

October 2020