

Piotr Tuzowski

**OBIKTOWO-ZORIENTOWANE ŚRODOWISKO
OBLICZENIOWE DLA ZAGADNIEŃ MECHANIKI
KONSTRUKCJI**



INSTYTUT PODSTAWOWYCH PROBLEMÓW TECHNIKI
POLSKIEJ AKADEMII NAUK
WARSZAWA 2005

Redaktor Naczelny:

doc. dr hab. Zbigniew Kotulski

Promotor:

prof. dr hab. inż. Michał Kleiber

Recenzenci:

prof. dr hab. inż. Tadeusz Burczyński

prof. dr hab. inż. Tomasz Łodygowski

Praca wpłynęła do Redakcji 28 lipca 2005 r.

Rozprawa doktorska

Instytut Podstawowych Problemów Techniki PAN

Nakład: 100 egz. Ark. Wyd. 9

Oddano do druku w październiku 2005 roku

Druk i oprawa: Drukarnia Braci Grodzickich, Piaseczno, ul. Geodetów 47a

Spis treści

1	Wstęp	6
1.1	Cel, przedmiot i zakres pracy	6
2	Podjęcie obiektowe w analizie MES, niezawodności i optymalizacji	10
2.1	Wstęp	10
2.2	Idea podejścia obiektowego	10
2.3	Wybór języka programowania	12
2.4	Podstawowe pojęcia podejścia obiektowego	12
2.4.1	Uwagi o terminologii	12
2.4.2	Analiza, projektowanie i programowanie obiektowe	13
2.4.3	Pojęcie klasy i obiektu	14
2.4.4	Struktury obiektowe	16
2.4.4.1	Struktura generalizacja-specjalizacja	16
2.4.4.2	Struktura całość-część	18
2.4.4.3	Komunikacja obiektów	19
2.5	Zastosowanie struktur obiektowych do stworzenia systemu analizy VFEM++	19
2.5.1	Struktura ogólna	19
2.5.2	Klasy elementów skończonych	20
2.5.3	Aproksymacja i geometria elementów skończonych	22
2.5.4	Klasy materiałów	23
2.5.5	Obiektowy model analizy skończenie elementowej	25
2.5.6	Klasy obciążeń elementowych	27
2.5.7	Klasy obiektów geometrycznych i generatora siatek regularnych	28
2.5.8	Graficzny interfejs użytkownika	30
2.5.9	Klasy analizy niezawodnościowej	31
2.5.10	Struktury klas optymalizacji	34
3	Zagadnienia termolepkosprężystości	38
3.1	Wstęp	38
3.2	O reologii	38

3.3	Niestacjonarny przepływ ciepła	39
3.3.1	Równanie przepływu ciepła	39
3.3.2	Sformułowanie MES problemu niestacjonarnego przepływu ciepła	39
3.4	Termolepkosprężystość	41
3.4.1	Model konstytutywny	41
3.4.2	Sformułowanie MES	44
3.5	Model zniszczenia	48
3.5.1	Wstęp	48
3.5.2	Koncepcja modelu zniszczenia	49
3.5.3	Model zniszczenia nielokalnego	54
3.5.4	Model zniszczenia w sformułowaniu skończenie elementowym	55
3.5.5	Termolepkosprężystość ze zniszczeniem	56
3.6	Przykład numeryczny - zniszczenie wspornika lepkosprężystego	61
3.7	Przykład numeryczny - lokalne i nielocalne zniszczenie belki lepkosprężystej	63
4	Analiza wrażliwości dla zagadnień termolepkosprężystych	74
4.1	Podstawowe pojęcia	74
4.1.1	Metody obliczania wrażliwości	75
4.1.1.1	Metoda różnic skończonych	75
4.1.1.2	Metoda bezpośredniego różniczkowania	75
4.1.1.3	Metoda układu sprzężonego	76
4.2	Analiza wrażliwości w problemie przepływu ciepła	76
4.3	Analiza wrażliwości w lepkosprężystości	77
4.4	Wrażliwość na parametry kształtu	80
4.5	Wrażliwość parametryczna w modelu zniszczenia	83
4.5.1	Wrażliwość parametryczna w modelu termolepkosprężystym ze zniszczeniem	83
4.5.2	Wrażliwość na parametry kształtu w modelu ze zniszczeniem	84
4.6	Przykład numeryczny - relaksacja naprężeń w pręcie	88
4.7	Przykład numeryczny - optymalizacja kształtu wspornika lepkosprężystego	92
5	Ocena niezawodności konstrukcji	96
5.1	Informacje wstępne	96
5.2	Charakterystyka metod szacowania niezawodności	97
5.2.1	Metoda Monte Carlo	97
5.2.2	First Order Reliability Method (FORM)	98
5.2.3	Second Order Reliability Method (SORM)	99
5.2.4	Importance sampling	102
5.3	Przykład numeryczny - analiza niezawodności pręta termolepkosprężystego	104
6	Generacja siatki i poprawianie dokładności obliczeń MES	108
6.1	Wstęp	108
6.2	Siatki regularne	109
6.3	Triangularyzacja parametryczna Delaunaya	113
6.3.1	Triangularyzacja właściwa	114
6.4	Oszacowanie błędu	116
6.5	Estymator błędu Zienkiewicza-Zhu	117

6.6	Wskaźnik błędu	118
6.7	Techniki adaptacyjne	118
6.8	Poprawianie siatki regularnej przez zagęszczenie	119
6.8.1	Aproksymacja związana	119
6.9	Przykłady numeryczne	122
6.9.1	Przykład numeryczny - test relaksacji	123
6.9.2	Przykład numeryczny - Połprzestrzeń sprężysta	125
6.9.3	Zagęszczanie siatek przez ponowną generację siatki	128
7	Wnioski	132

1.1 Cel, przedmiot i zakres pracy

Metoda elementów skończonych (MES) jest już od kilku dziesięcioleci potężnym i bardzo szeroko stosowanym narzędziem umożliwiającym inżynierom i naukowcom wykonywanie różnorodnych analiz. Najogólniej rzecz ujmując MES jest metodą umożliwiającą przybliżone rozwiązywanie równań różniczkowych. Stąd tak wszechstronne zastosowanie w nauce i technice. Obszar zainteresowań autora ogranicza się do zastosowań MES w termomechanice. Wszechstronność tej metody zaowocowała istnieniem zarówno wielu dużych, komercyjnych systemów o różnorodnych możliwościach jak i mniejszych, wyspecjalizowanych programów obliczeniowych. W wielu ośrodkach naukowych tworzone są programy metody elementów skończonych, służące do przeprowadzania eksperymentów numerycznych testujących nowe idee rozwijającej się bardzo dynamicznie dziedziny, jaką jest skończenie elementowa analiza materiałów i konstrukcji. Istotnym aspektem projektowania konstrukcji jest poszukiwanie rozwiązań bardziej optymalnych i niezawodnych, dlatego też duże znaczenie ma analiza optymalizacyjna, niezawodnościowa oraz optymalizacja niezawodnościowa. Programy komputerowe optymalizacji i niezawodności bazują na wynikach dostarczanych przez programy MES. Stąd ważnym kierunkiem rozwoju wydaje się być integracja w jednym programie procedur analizy MES i optymalizacji konstrukcji, co jest jednym z celów tej pracy.

Złożony system obliczeniowy MES jest programem komputerowym umożliwiającym analizowanie wielu różnych rodzajów konstrukcji takich jak belki, ramy, tarcze w płaskim stanie odkształcenia czy naprężenia, konstrukcje trójwymiarowe w trójosiowym stanie naprężenia. W systemach tych istnieje możliwość przeprowadzania rozmaitych typów obliczeń np. statycznych, dynamicznych czy przepływu ciepła. Najmniejsze z programów MES składają się zwykle z kilku tysięcy linii kodu. System stworzony w ramach tej pracy składa się z ponad dziesięciu tysięcy linii kodu. Kod źródłowy komercyjnych systemów MES składa się nierzadko z kilkuset tysięcy linii. Postęp w dziedzinie komputeryzacji, lawinowo rosnąca moc obliczeniowa współczesnych komputerów sprawia, że rośnie ape-

tyt na rozwiązywanie coraz bardziej skomplikowanych zadań. Rosnący stopień złożoności systemów MES oraz wymagane przez dzisiejszy rynek szybkie tempo prac stanowi duże wyzwanie dla twórców pracujących nad tworzeniem i rozwijaniem budowanych przez nich systemów. Uwarunkowania te wymagają zastosowania wydajnych narzędzi programistycznych do projektowania i budowy współczesnych programów a stopień komplikacji sprawia, że narzędzia powinny zapewniać łatwość budowy i testowania programów. Tym trudnym wyzwaniom potrafi sprostać podejście obiektowe. Tradycyjne podejście strukturalne stosowane często w przeszłości nie dostarcza efektywnych narzędzi pozwalających na łatwe opisywanie złożonych systemów tak, jak to czyni od czterech dekad przeżywające obecnie rozkwit podejście obiektowe. Do budowy programu z zastosowaniem paradygmatu programowania strukturalnego potrzeba więcej czasu i ludzi. Stopień złożoności systemu skończenie elementowego jest na tyle duży, że podejście obiektowe wydaje się być najlepszym rozwiązaniem. Współczesne systemy MES wyposażone są zwykle w pre i postprocesory pozwalające na graficzne zobrazowanie rezultatów obliczeń w postaci map warstwicznych czy animacji. Programowanie obiektowe ma również duże zastosowanie przy budowie takich graficznych programów wizualizacyjnych. Testowanie zalet podejścia obiektowego jest jednym z celów tej pracy. W jej ramach stworzony został system MES VMES++, wykorzystujący techniki obiektowe do analizy konstrukcji.

Architektura systemu pozwala nazwać go środowiskiem obliczeniowym. Środowisko obliczeniowe to zbiór modułów umożliwiających realizację bardzo różnorodnych eksperymentów numerycznych. Ważną cechą tego środowiska jest możliwość jego rozbudowy. Zastosowanie do stworzenia systemu paradygmatu programowania obiektowego pozwala na relatywnie łatwe dodawanie nowych funkcji - od poszerzania spektrum możliwości w zakresie analiz MES, niezawodności, czy optymalizacji - do usprawnienia pewnych aspektów technicznych jak np.: przestrzenna wizualizacja obliczeń z wykorzystaniem akceleratorów graficznych i języka OpenGL. Najważniejszymi elementami środowiska VFEM++ są: moduł analizy MES, moduł analizy optymalizacyjnej, moduł analizy niezawodnościowej, czy moduł postprocesora umożliwiający wizualizację rezultatów obliczeń.

Istotnym aspektem projektowania współczesnych konstrukcji inżynierskich jest szukanie rozwiązań najbardziej optymalnych. Wśród wielu metod optymalizacji konstrukcji, metody gradientowe są metodami najbardziej wydajnymi. Podobnie w analizie niezawodnościowej, najefektywniejszymi metodami oszacowania niezawodności są gradientowe metody FORM i SORM. Dla zastosowania metod gradientowych system MES musi posiadać możliwość obliczania gradientów funkcji celu, gradientów ograniczeń, czy w niezawodności, gradientów funkcji granicznej. Obliczaniem takich gradientów zajmuje się analiza wrażliwości. Będący częścią tej pracy system MES posiada funkcje analizy wrażliwości, umożliwiającą obliczanie gradientów potrzebnych w optymalizacji i niezawodności wielkości.

Kolejnym istotnym problemem analizy skończenie elementowej jest analiza dokładności obliczeń. Jak już wspomniano, metoda elementów skończonych jest metodą przybliżoną, zatem naturalnym problemem jest kwestia dokładności otrzymywanych rezultatów. Twierdzenia o zbieżności określają warunki, spełnienie których gwarantuje zbieżność rezultatów do rozwiązania dokładnego, wraz ze zwiększaniem liczby stopni swobody modelu. Zagęszczanie siatki powoduje jednak zwiększanie kosztów obliczeń. Dlatego też siatkę należy zagęszczać selektywnie w tych obszarach konstrukcji, w których błąd obliczeń nie mieści się w zadanej przez użytkownika tolerancji. Istnieją służące do szacowania błędu obliczeń MES estymatory błędu (ang. *error estimator*) pozwalające określić błąd obliczeń

i na podstawie jego wartości dokonać poprawienia jakości siatki (ang. *mesh refinement*) selektywnie. System VFEM++ posiada możliwość stosowania technik adaptacyjnych.

W skład pracy wchodzi Wstęp, siedem Rozdziałów oraz Wnioski i Bibliografia.

Drugi rozdział poświęcony jest podejściu obiektowemu w zastosowaniu do budowy systemu metody elementów skończonych. Na wstępie rozdziału zaprezentowane zostały idee zastosowania podejścia obiektowego oraz przedstawione podstawowe pojęcia. Następnie pokazano sposób opisu podstawowych pojęć z dziedziny MES takich, jak: węzeł, element skończony, materiał czy aproksymacja pola przemieszczeń. Na schematach graficznych przedstawiono struktury danych odzwierciedlające wzajemną współpracę w/w obiektów będącą jednym z kluczowych aspektów podejścia obiektowego.

Problematyka trzeciego rozdziału obejmuje zagadnienia związane z analizą MES konstrukcji termolepkosprężystych. Po krótkim wstępie, poświęconym reologii, przedstawiony został konstytutywny model materiału termolepkosprężystego wraz z jego krótką charakterystyką. Następnie wyprowadzono równania MES dla tego modelu oraz przedstawiono sformułowanie problemu niestacjonarnego przepływu ciepła. W rozdziale tym przedstawiono też sposób modelowania zniszczenia oraz koncepcje nielokalnego sformułowania zniszczenia, znacznie ograniczającego wpływ siatki MES na rezultaty co czyni obliczeń bardziej obiektywnymi. Następnie wyprowadzono równania MES dla termolepkosprężystego modelu materiału, z uwzględnieniem zniszczenia.

Czwarty rozdział zawiera opis zastosowania analizy wrażliwości do opisanego w poprzednim rozdziale modelu termolepkosprężystego. W części wstępnej zaprezentowane zostały podstawowe pojęcia analizy wrażliwości oraz zwięźle opisane metody obliczania wrażliwości. Następnie wyprowadzono równania MES dla obliczania wrażliwości problemu niestacjonarnego przepływu ciepła oraz równania wrażliwości dla termolepkosprężystego modelu materiału, a także podano algorytm obliczania wrażliwości. Ponadto dla obu ww. zagadnień wyprowadzono równania MES dla wrażliwości na parametry kształtu.

Piąty rozdział poświęcony jest analizie niezawodności. W rozdziale tym przedstawiono na wstępie cele analizy niezawodności jako metody obliczania prawdopodobieństwa awarii konstrukcji. Następnie opisane zostały podstawowe metody stosowane w analizie niezawodności konstrukcji takie, jak: Metoda Monte Carlo, FORM, SORM, oraz Importance Sampling.

W szóstym rozdziale zawarto opis technik poprawiania jakości siatki MES w oparciu o estymator błędu. Na wstępie zaprezentowano różne rodzaje siatek regularnych i nieregularnych, a następnie przedstawiono ideę oraz podstawowe wzory dla estymatora błędu. W dalszej części rozdziału przedstawiono techniki adaptacyjne dla siatek regularnych i nieregularnych (uzyskanych przez triangularyzację) oraz przedstawiono algorytm triangularyzacji Delaunaya.

Zamykający pracę rozdział siódmy zawiera wnioski i spostrzeżenia wynikające z otrzymanych w pracy rezultatów.

Podjęcie obiektowe w analizie MES, niezawodności i optymalizacji

2.1 Wstęp

W tym rozdziale przedstawiono zastosowanie podejścia obiektowego do programowania skończenie elementowego. Na wstępie przedstawione zostaną podstawowe pojęcia z bardzo szerokiej dziedziny, jaką są metody obiektowe w informatyce, [8], [5]–[7]. Główny nacisk położony zostanie na zastosowanie podejścia obiektowego w metodzie elementów skończonych. Przedstawione zostaną struktury danych zastosowane w obiektowo zorientowanym środowisku obliczeniowym VFEM++ będącym częścią tej pracy. Zaprezentowane i omówione będą najistotniejsze obiektowe struktury danych i powiązania między nimi składające się na opisywany system.

Już ponad dziesięć lat tematyka obiektowa obecna jest na łamach wiodących czasopism naukowych. Pierwszą znaczącą pracę przedstawiającą architekturę programu analizy konstrukcji zaprezentował Miller (1990) [14]. Od tego czasu sukcesywnie rosło zainteresowanie podejściem obiektowym w zastosowaniu do budowy systemów MES. Zaowocowało to wieloma publikacjami [11]–[17],[21]–[29]. Również polscy naukowcy interesują się tą dziedziną [18]–[20]. Pojawiły się też interesujące pozycje książkowe na ten temat por. [30, 33].

2.2 Idea podejścia obiektowego

Podstawą do stworzenia każdego systemu jest dobre zrozumienie modelowanego problemu. Następnie trzeba zastosować efektywne metody do opisu problemu. Wobec lawinowo rosnącej złożoności systemów MES, bardziej tradycyjne podejście strukturalne wydaje się być mało wydajne. Paradygmat podejścia strukturalnego brzmi: [4]:

Zadecyduj, jakie chcesz mieć procedury, stosuj najlepsze algorytmy, jakie możesz znaleźć

Z tego paradygmatu wynika, że jednym z najważniejszych pojęć jest algorytm. Na dziedzinę problemu patrzy się zatem jak na zbiór czynności do wykonania zaniehbując niejako struktury danych. Z czasem zorientowano się, że istotną rolę w procesie budowy systemów informatycznych odgrywa organizacja danych. Dostrzeżono związek pomiędzy procedurami a strukturami danych. Powstał więc bardziej zaawansowany paradygmat programowania modularnego mianowicie: [4]

Zadecyduj, jakie chcesz mieć moduły, podziel program w taki sposób, aby ukryć dane w modułach.

Paradygmat stosowany m.in. w języku FORTRAN 90/95. Program podzielony jest na części zwane modułami, a każdy moduł zawiera deklaracje struktur danych oraz pojęciowo z nimi związane procedury. Pozwala to na budowanie lepszych modeli niż w przypadku podejścia strukturalnego i jest krokiem w kierunku podejścia obiektowego znacznie rozwijającego ideę łączenia danych i procedur. Paradygmat programowania obiektowego brzmi: [4]

Zadecyduj, jakie chcesz mieć klasy; dla każdej klasy dostarcz pełny zbiór operacji; korzystając z mechanizmów dziedziczenia jawnie wskaż to, co wspólne.

Według tego paradygmatu stworzenie odpowiednich struktur danych i określenie relacji między nimi odgrywa kluczową rolę. Dzięki temu komputerowy model stworzony za pomocą takich struktur jest znacznie bliższy problemowi, bo niemal wprost odzwierciedla rzeczywiste zależności pomiędzy elementami składowymi rozpatrywanej dziedziny problemu. Większość dziedzin życia a w szczególności różne dziedziny nauki i techniki posługują się klasyfikacją jako podstawową metodą ustanowienia porządku w pojęciach składających się na nią. Wprowadza to przejrzystą strukturę i pozwala zdecydowanie prościej poruszać się w danej dziedzinie. Na przykład trudno sobie wyobrazić matematykę bez istnienia pojęć typu: ciało, grupa, przestrzeń liniowa, czy innych, podobnych wzajemnie powiązanych struktur. W rozpatrywanym w tej pracy zagadnieniu - metodzie elementów skończonych - można z łatwością stworzyć podobne klasyfikacje. Na przykład elementy skończone można podzielić na jedno-, dwu- i trójwymiarowe. Można wyróżnić klasę elementów izoparametrycznych. Każdy typ elementu skończonego może mieć wiele typów materiałów, konstrukcje można poddawać różnym rodzajom analiz. Dzięki mechanizmom obiektowym tworzymy struktury danych niemal bezpośrednio odzwierciedlające te klasyfikacje naszego problemu. Deklarowane w obiektowym języku programowania struktury danych, zwane klasami, są właśnie tym obrazem naturalnych elementów klasyfikacji. Dzięki tej naturalności budowa struktur danych jest znacznie prostsza, niż w przypadku programowania strukturalnego, gdzie struktury danych nie mają wiele wspólnego z naturalną klasyfikacją. Czytelniejszy model pozwala jednej osobie na kontrolę znacznie bardziej złożonych problemów, niż w podejściu strukturalnym. Sprawia, że system jest łatwiejszy w utrzymaniu i prostszy w rozbudowie. W podejściu strukturalnym często łatwiej jest napisać program od nowa, niż go zmodyfikować. Podejście obiektowe daje również możliwości wyrażania związków pomiędzy poszczególnymi klasami, czyli tworzenia struktur obiektowych, co będzie opisane w dalszej części rozdziału.

2.3 Wybór języka programowania

Najbardziej rozpowszechnionym językiem programowania aplikacji mechanicznych jest FORTRAN. Choć powstają coraz to nowsze implementacje tego języka, ciągle nie daje on wsparcia programistom stosującym podejście obiektowe. Wśród języków wspierających techniki obiektowe najbardziej rozpowszechnionym jest niewątpliwie C++. Na większości platform sprzętowych i systemów operacyjnych istnieje nierzadko kilka kompilatorów tego języka. W wielu z nich istnieją bardzo wygodne środowiska programistyczne, które zawierają nie tylko kompilator, ale i szereg innych pożytecznych narzędzi do konstrukcji oprogramowania. Język C++, oprócz technik obiektowych, daje inne mechanizmy bardzo pożyteczne przy programowaniu skończenie elementowym. Możliwe jest definiowanie operatorów arytmetycznych przez użytkownika, co sprawia, że wyrażenia macierzowe zapisuje się jak w matematyce z użyciem operatorów. Program w języku C++ jest zatem bardzo czytelny, co nie jest bez znaczenia w przypadku skomplikowanego systemu MES. Podobne, choć nie tak uniwersalne, mechanizmy wprowadzono w FORTRANIE 90/95: [4]

“C++ zaprojektowano tak, by umożliwić racjonalną strukturalizację dużych programów, by pojedyncza osoba była w stanie uruchomić 25000 wierszy kodu. Dotychczasowa praktyka pokazuje, że C++ wypełnia to zadanie z nawiązką”

Potwierdza to również ta praca, gdyż stworzony przez autora system VFEM++ jest obiektowym programem składającym się z ponad 10000 linii kodu. Uruchomienie i rozszerzanie programu, według opinii autora, nie sprawia zbyt wielu kłopotów. Wpływ na wybór języka mogą mieć preferencje autorów programów, a w szczególności zdobyte w pracy zawodowej doświadczenie. Dalsze rozważania na temat podejścia obiektowego, zaprezentowane w kolejnych punktach pracy, mają charakter ogólny i dotyczą każdego języka implementującego techniki obiektowe. Zapowiadana wersja FORTRAN 2003 ma już wspierać podejście obiektowe.

2.4 Podstawowe pojęcia podejścia obiektowego

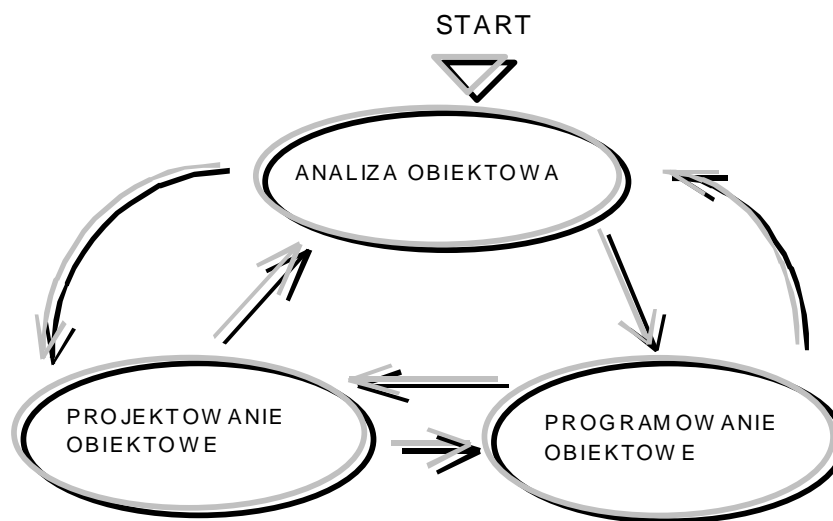
2.4.1 Uwagi o terminologii

Tak, jak w każdej dziedzinie, również w programowaniu obiektowym istnieje specyficzna terminologia. Niestety nie jest ona ujednolicona. Różni autorzy wprowadzają swoje własne nazewnictwo, zwykle podobne, ale często nieco odmienne od innych. Są też różne podejścia do modelowania struktur obiektów. Diagramy struktur obiektowych przedstawiane są w różny sposób przez różnych autorów. Istnieją prace będące porównaniem podejść do metod analizy i projektowania obiektowego [3]. Martin [8] podaje tabelaryczne zestawienie podstawowych terminów podejścia obiektowego stosowanego przez czołowych autorów prac poświęconych obiektowości. Na szczęście nie ma większego znaczenia, jaka konwencja nazewnictwa zostanie przyjęta. Powodem wyboru tej czy innej graficznej reprezentacji struktur obiektowych może być to, jakie informacje zawarte są w diagramach. W tej pracy zastosowany będzie sposób Coad'a i Yourdona [5]-[7].

2.4.2 Analiza, projektowanie i programowanie obiektowe

Techniki obiektowe dają narzędzia do modelowania bardzo skomplikowanych problemów. Im bardziej złożony problem, tym bardziej skomplikowany jest również proces tworzenia oprogramowania modelującego ten problem. Wobec tych wyzwań konieczne stało się wypracowanie technik efektywnego tworzenia zaawansowanego oprogramowania. Proces tworzenia programu obiektowego składa się z trzech części:

- analiza obiektowa
- projektowanie obiektowe
- programowanie obiektowe



Rysunek 2.1: Schemat procesu tworzenia aplikacji obiektowej

Analiza Obiektowa - to pierwszy etap tworzenia programu. Jednym z zasadniczych elementów tej fazy jest dogłębne zrozumienie problemu oraz określenie obowiązków systemu czyli zadań jakie musi spełniać tworzony program. Gruntowna znajomość dziedziny i zadań systemu pozwala następnie na stworzenie zbioru klas, atrybutów i metod będących składowymi modelowanego zagadnienia. Wykorzystuje się do tego programy narzędziowe wspomagające analizę obiektową. Nie określa się w tej fazie sposobów realizacji zadań realizowanych przez obiekty. Czynności na tym etapie należy jedynie wyspecyfikować. Na przykład wiadomo, że obiekt opisujący macierz posiada funkcję dekompozycji macierzy. Na etapie analizy nie określa się, jaką metodą będzie ona dekomponowana.

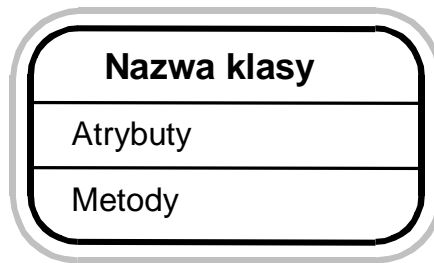
Projektowanie Obiektowe - na tym etapie na podstawie wcześniejszych obserwacji analitycznych tworzona jest konstrukcja systemu, z uwzględnieniem szczegółów związanych z implementacją komputerową. Tutaj, w odróżnieniu od analizy, istotny jest aspekt komputerowej realizacji zadania. Na tym etapie uzupełnia się np. opis klas o specyfikację

składowych (rozdzielanie składowych na publiczne i składowe). Należy też zaprojektować interfejs komunikacji z użytkownikiem.

Programowanie obiektowe - kodowanie programu w danym języku programowania na bazie stworzonego wcześniej projektu.

Wymienione powyżej etapy nie są jednak realizowane ściśle jeden po drugim. Często powraca się do poprzednich etapów. Czasami na etapie analizy nie da się przewidzieć wszystkich występujących zjawisk lub wymagań narzucanych podczas programowania. Często trzeba stworzyć programy prototypowe, aby przekonać się o słuszności wcześniejszych założeń. W procesie tworzenia programu możliwe jest więc wielokrotne powtarzanie poszczególnych etapów. Schematycznie ten iteracyjny proces zobrazowany jest na Rys. 2.1. Podane etapy tworzenia programu obiektowego stanowią dość złożony proces. Szczegółową wiedzę na temat ich realizacji można znaleźć w bogatej literaturze [5]-[7],[8],[4],[10].

2.4.3 Pojęcie klasy i obiektu



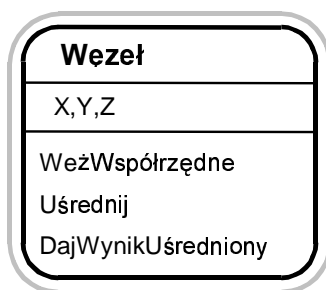
Rysunek 2.2: Przykład diagramu obrazującego klasę

Podstawowym pojęciem podejścia obiektowego jest *obiekt*. Formalnie obiekt jest zmienną zadeklarowaną w danym języku programowania. Jest to zmienna typu rekordowego zawierająca zbiór innych zmiennych i procedur zwanych *składowymi*. Taka deklaracja budowy obiektu nazywa się *klasą*. Inaczej mówiąc *klasa* jest typem danych, a utworzona zmienna tego typu jest obiektem. W terminologii obiektowej taką zmienną nazywa się *egzemplarzem* (ang. *instance*). Często jednak zdarza się, że pojęcia klasy oraz obiektu traktowane są jak synonimy. Wymienne mówi się o składowych klasy lub obiektu. Tworzenie klas nie jest czynnością formalną, więc nazewnictwo w programowaniu obiektowym odzwierciedla sens, jaki mają poszczególne pojęcia. Dlatego niektóre z nazw są inne niż te, do których jest się przyzwyczajonym programując np. strukturalnie, (funkcja, procedura, zmienna). Zmienne, składowe klasy, nazywa się *atrybutami* lub *cechami* obiektu, gdyż opisują one istotne dla tego obiektu właściwości. Procedury i funkcje natomiast nazywa się *metodami* lub *usługami*. Realizują one wszystkie czynności przypisane temu obiektowi i tylko one mogą mieć wyłączny dostęp do atrybutów i mogą je przetwarzać. Schematycznie na diagramach klasy będą przedstawiane tak, jak na Rys. 2.2. Przedstawiony jest na nim schemat klasy z jej nazwą, atrybutami oraz metodami.

Są takie klasy, które służą tylko do deklaracji struktur obiektowych. Z tych klas nigdy nie powstają egzemplarze. Takie klasy nazywamy klasami *abstrakcyjnymi*. Ich zastosowanie będzie szerzej omówione w następnych punktach. Natomiast klasy nieabstrakcyjne, czyli takie z których tworzy się w programie obiekty, noszą nazwę *klasy-i-obiektu*. Szara

obwódka na schemacie (Rys. 2.2) oznacza że jest to klasa-i-obiekt, natomiast jej brak oznacza klasę abstrakcyjną.

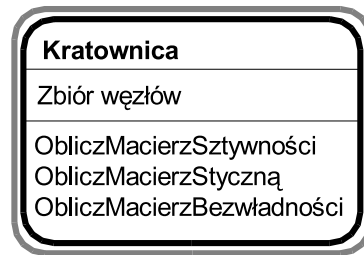
Deklaracje takich struktur oraz możliwość tworzenia obiektów są potężnymi narzędziami doskonale wspierającymi modelowanie rzeczywistych problemów. Klasa jest strukturą reprezentującą bowiem określone w dziedzinie problemu pojęcie, przechowując w sobie wszystkie dane potrzebne do opisu danego zjawiska i definiujące wszystkie operacje z pojęciem tym związane, a udostępniając “na zewnątrz” tylko wybrane rezultaty działania i ukrywając wszystkie nieistotne dla przyszłego użytkownika obiektu szczegóły działania. To postępowanie nazywa się *hermetyzacją* danych. Dzięki temu mamy pewność, że dana czynność jest wykonywana tam, i tylko tam gdzie powinna być wykonywana, co znakomicie ułatwia szukanie błędów w programie oraz pozwala na rozszerzanie i modyfikację programu. Wystarczy bowiem tylko zmodyfikować deklarację tego obiektu (klasę). Wiadomo, że wszystkie dane i czynności związane z tym pojęciem są tam zadeklarowane. To tylko pozornie przypomina postępowanie w podejściu strukturalnym bo deklaracje procedur nie mają żadnego formalnego powiązania z deklaracjami zmiennych. W obiekcie właśnie takie formalne powiązanie danych i przetwarzających je procedur ma miejsce i określa się je terminem *abstrakcja danych*. Takie podejście pozwala znacznie naturalniej opisać zjawiska występujące w rzeczywistości. Klasa bardziej bezpośrednio modeluje rzeczywiste obiekty. Dobrym przykładem jest urządzenie elektroniczne. Każda część jest obiektem wykonującym pewne czynności na rzecz urządzenia, a jej cechami są parametry (np. w kondensatorze pojemność). Urządzenie to jest zamknięte w obudowie i komunikuje się ze światem zewnętrznym tylko poprzez wychodzące z niego przewody a jego działanie jest ukryte pod obudową i niedostępne. Owe przewody to, w przypadku obiektów, procedury dostępne dla użytkownika, dzięki którym obiekt komunikuje się ze światem zewnętrznym, a wartości funkcji składowych to rezultaty jego działania.



Rysunek 2.3: Klasa węzeł

W interesującej nas dziedzinie, jaką jest analiza konstrukcji metodą elementów skończonych, można bez trudu dostrzec wiele podobszarów tej dziedziny, które mogą (a nawet, można by rzec, aż się o to proszą) być modelowane przy pomocy klas. Najprostszym przykładem może być struktura danych opisująca węzeł. W klasie Węzeł (Rys. 2.3) widzimy atrybuty w postaci współrzędnych węzła (są to trzy liczby rzeczywiste X, Y, Z) i trzy składowe funkcje: jedna zwracająca macierz współrzędnych węzła *WeżWspółrzędne* i dwie funkcje służące do uśredniania wyników w węzłach.

Drugim przykładem jest (Rys. 2.4) klasa elementu skończonego kratownicy. Posiada ona zbiór węzłów, na których “rozpięty” jest element oraz funkcje obliczające macierz



Rysunek 2.4: Klasa Kratownica

szywności, macierz bezładności i macierz styczną. Liczba zmiennych obiektowych (egzemplarzy) będzie taka jak odpowiednia ilość węzłów i elementów konstrukcji analizowanego zadania. Każdy egzemplarz przechowuje swoje dane (np. współrzędne). Przykładowe klasy zamieszczone w tym rozdziale ilustrują jedynie omawiane mechanizmy obiektowe. Klasy składające się na tworzony w ramach tej pracy VFEM++ posiadają znacznie więcej funkcji składowych (obliczanie macierzy transformacji, macierzy kształtu, tablicowania wyników), które nie są prezentowane na diagramach by nie przesłaniać istoty problemu.

2.4.4 Struktury obiektowe

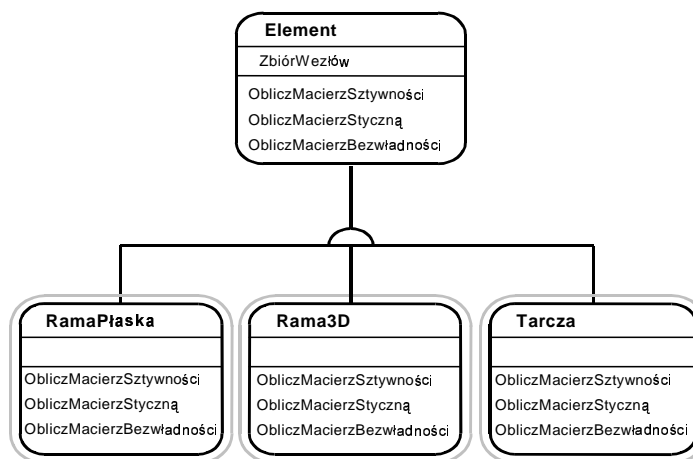
Nietrudno zauważyć, że klasyfikacja pojęć rozpatrywanej dziedziny wskazuje wspólne cechy i różnice. Istotna jest więc możliwość odzwierciedlenia tych naturalnych klasyfikacji w modelu obiektowym. Dla przykładu rozważmy podobieństwa elementów skończonych. Cechą wspólną wszystkich elementów skończonych jest to, że posiadają niepusty zbiór węzłów, na których są rozpięte. Niektóre elementy mają ze sobą więcej cech wspólnych niż inne. Na przykład element ramy płaskiej ma więcej cech wspólnych z elementem ramy przestrzennej niż z elementem powłokowym. Znajdowanie tych cech wspólnych i możliwość efektywnego wykorzystania ich jest jedną z ważniejszych właściwości podejścia obiektowego. Takie znajdowanie bardziej ogólnych typów nosi nazwę uogólniania. Widać to nawet w paradygmacie programowania obiektowego (punkt 2.2). W podejściu strukturalnym języki programowania nie dawały żadnego wsparcia, żadnych mechanizmów pozwalających wykorzystać cechy wspólne pomiędzy elementami skończonymi. Twórcy oprogramowania musieli sami rozwiązywać te problemy. Wprowadzało to dodatkowe komplikacje. Podejście obiektowe poprzez stworzenie odpowiednich relacji pomiędzy obiektami daje nam możliwość efektywnego wykorzystania cech wspólnych, a przez to dużego uproszczenia procesu tworzenia nawet bardzo złożonych struktur.

2.4.4.1 Struktura generalizacja-specjalizacja

Jednym z typów powiązań obiektów jest struktura generalizacja-specjalizacja (w skrócie *gen-spec*). Struktura ta realizowana jest przy pomocy mechanizmu dziedziczenia. Dziedziczenie oznacza przekazywanie cech jednej klasy klasie dziedziczącej. Dziedzicząca klasa będzie posiadała atrybuty i usługi klasy, z której dziedziczy, mając możliwość dodania swoich atrybutów lub usług. Takie powiązanie klas poprzez dziedziczenie nazywa się hierarchią klas. Przykładowa hierarchia klas elementów skończonych pokazana jest na Rys. 2.5 Klasa Element posiada atrybuty ZbiórWęzłów i dziedziczące z niej klasy RamaPłaska,

Rama3D oraz Tarcza będą również posiadały ten atrybut, czyli nie trzeba go już w klasie deklarować.

Ważnym mechanizmem poszerzającym funkcjonalność dziedziczenia i wspierającym opis wzajemnych relacji klas jest *polimorfizm*. Polimorfizm umożliwia dziedziczącym klasom posiadanie własnych definicji danej funkcji zadeklarowanej w klasie bazowej. Takie polimorficzne funkcje nazywa się funkcjami *wirtualnymi*. Jest to również zobrazowane na Rys. 2.5. W Klasie bazowej **Element** zadeklarowano funkcje: **MacierzSztynności**, **MacierzStyczną** i **MacierzBezwności**. Dziedziczące z niej klasy również posiadają deklarację tych funkcji, przy czym treść tych funkcji w każdej z klas jest inna. Abstrakcyjna klasa **Element** istnieje tylko po to, aby zapostulować pewien zestaw cech, które musi posiadać element skończony. Zatem w klasie **Element** funkcje są również abstrakcyjne tzn. nie posiadają ciała. Treść tych funkcji jest zdefiniowana w dziedziczących z niej klasach-i-objektach reprezentujących konkretne elementy skończone. Takie hierarchie dziedziczenia mogą być bardziej rozbudowane. Hierarchia może być rozbudowana "wzdłuż", kiedy wiele klas dziedziczy jedna z drugiej lub, kiedy jedna klasa dziedziczy z kilku klas. Półkola na schemacie, Rys. 2.5, są oznaczeniem struktury gen-spec, czyli dziedziczenia.

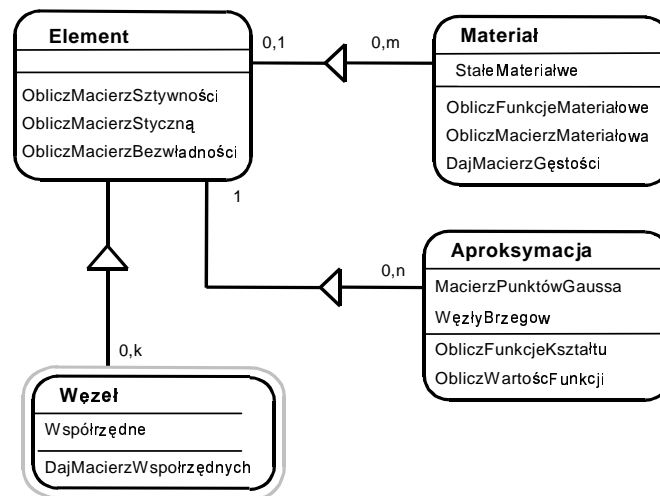


Rysunek 2.5: Klasa Kratownica3D

Widać tu ogromną rolę, jaką odgrywają dziedziczenie i polimorfizm. To właśnie te mechanizmy dają możliwość budowania złożonych struktur efektywnie wykorzystując wspólne cechy klas. Pozwalają zdefiniować pewne klasy na dużym poziomie ogólności, postulując istnienie atrybutów i funkcji (generalizacja). Poprzez dziedziczenie budowana jest hierarchia, na końcu której znajdują się konkretne klasy-i-objekty (specjalizacja). W programowaniu strukturalnym nie ma o tym mowy. Program zawiera tylko konkretny kod realizujący konkretny algorytm. W podejściu obiektowym definiujemy struktury programu na znacznie większym poziomie ogólności. Dzięki klasom abstrakcyjnym można zbudować system klas będący podstawą do budowy programu i jednocześnie ułatwiający tworzenie konkretnych klas-i-objektów dla zagadnień stojących na niższym poziomie hierarchii dziedziczenia.

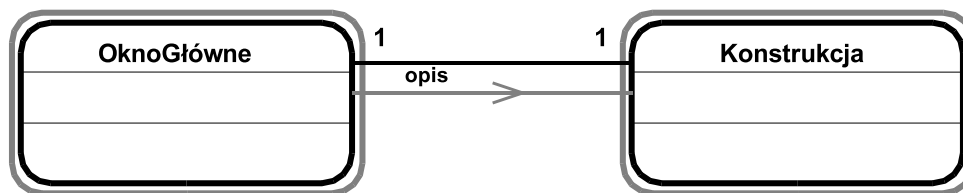
2.4.4.2 Struktura całość-część

Innym powiązaniem obiektów jest struktura całość-część. Przy pomocy tej struktury opisywane są pojęcia o takiej złożoności, w której główny obiekt reprezentujący dane pojęcie zawiera inne obiekty współpracujące z nim i będące w rzeczywistości jego częściami. Nie trudno pokazać praktyczne przykłady obiektów posiadających części. Na przykład samochód posiada silnik, czy koła. Podobnie w dziedzinie, jaką jest analiza konstrukcji metodą elementów skończonych, można znaleźć wiele pojęć, których składowe można by nazwać częściami. Przykładowo częściami elementu skończonego możemy nazwać materiał elementu, węzły, na których jest rozpięty czy aproksymacje pola wynikowego. Przykład struktury całość-część pokazany jest na Rys. 2.6. Na schemacie widzimy obiekt *Element*,



Rysunek 2.6: Przykład struktury całość część

który zawiera trzy inne obiekty. Kierunek zaznaczony na rysunku skierowany jest od obiektu-części do obiektu-całości. Liczby umieszczone na połączeniu oznaczają maksymalną liczbę posiadanych obiektów. Obiekt *Element* współpracuje z k węzłami. Pozornie może to wyglądać na sprzeczne z poprzednimi przykładami, gdzie *ZbiórWęzłów* był atrybutem obiektu. To, że *ZbiórWęzłów* jest atrybutem, jest tylko szczegółem realizacyjnym struktury całość-część. Obecnie rozpatrujemy sposób działania struktury, a nie szczegóły implementacyjne. Obiekt *Element* posiada najwyżej jeden materiał. Ten sam materiał natomiast może być przypisany do m elementów. Oczywiście brak materiału spowoduje błąd przy rozwiązaniu zadania, ale jest formalnie dopuszczalny przy wprowadzaniu danych zadania. Podobnie z obiektem *Aproksymacji*. Obiekt zawsze posiada jedną aproksymację (np. *Lagrange 16* węzłowa). Ta sama aproksymacja może być przypisana n obiektom. Zgodnie z pojęciami hermetyzacji i abstrakcji danych obiekt pobiera wszystkie informacje dotyczące materiału z obiektu *Material*, zaś informacje o funkcji kształtu i geometrii elementu z obiektu *Aproksymacja*, z którymi jest połączony w strukturze całość-część. W programie struktury całość-część wykorzystuje się łącznie ze strukturami gen-spec, co zostanie pokazane już na konkretnych przykładach struktur programu VFEM w dalszej części pracy.



Rysunek 2.7: Komunikacja obiektów

2.4.4.3 Komunikacja obiektów

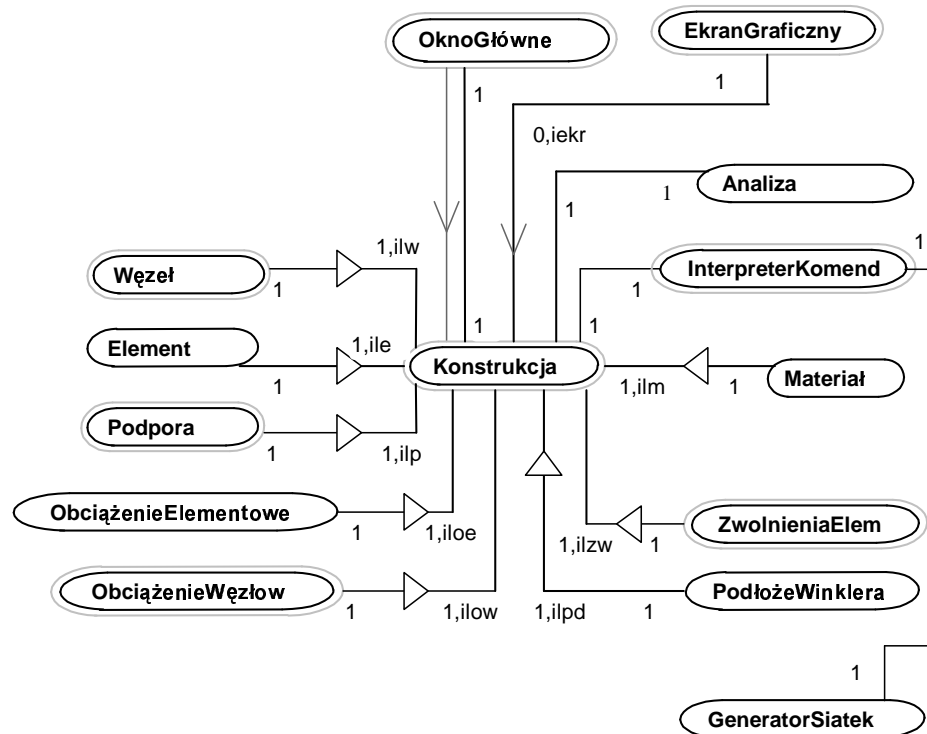
Powiązaniem nazywamy taki związek dwóch obiektów, w którym obiekty te nie mają żadnych cech wspólnych, żaden nie jest częścią drugiego, a mimo to ze sobą współpracują. Przykład powiązania znajduje się na Rys. 2.7. Obiekt **OknoGłówne** reprezentuje główne okno programu. (w okienkowym systemie operacyjnym np.: Windows). Natomiast **Konstrukcja** jest obiektem będącym w istocie bazą danych zawierającą wszelkie dane analizowanej konstrukcji. Obiekt **OknoGłówne** reprezentuje okno Windows. Odpowiada tylko za współpracę z użytkownikiem i nie ma nic wspólnego z analizą konstrukcji. Nie ma więc żadnych cech wspólnych z obiektem **Konstrukcja**. Nie da się zastosować struktury gen-spec ani struktury całość-część. Obiekty te muszą jednak współpracować ze sobą. Fizycznie ta współpraca odbywa się tak samo jak w strukturze całość-część tj. przez wzajemne wywoływanie składowych funkcji (pojęciowo okno programu nie jest przecież częścią konstrukcji ani konstrukcja częścią okna). Główne okno (wersji okienkowej programu) zawiera menu oraz przyciski sterujące, przy pomocy których użytkownik wywołuje poszczególne funkcje programu. Funkcje związane z konstrukcją takie, jak: wprowadzenie danych, wykonanie obliczeń, narysowanie schematu, wykonuje obiekt **Konstrukcja**. Stąd też konieczność powiązania tych obiektów i ich współpracy. Na Rys. 2.7 widać zaznaczone czarną linią powiązanie. Szarą, ukierunkowaną linią zaznacza się wykonywanie funkcji. Linia skierowana jest od obiektu wywołującego funkcję do obiektu, którego ta funkcja jest składową. Takie wywołanie funkcji nazywa się *komunikatem*. Nad strzałką można umieścić opis komunikatu. Często jednak chodzi o zaznaczenie tylko faktu współpracy obiektów i opisu nie umieszcza się. Mogą ze sobą współpracować obiekty powiązane strukturą całość-część. Oznaczenie komunikatu może być więc również pomiędzy obiektem-całością, a obiektem-częścią.

Pokazane na schematach struktury obiektowe są statycznym obrazem opisywanego problemu. Zaznaczenie na schemacie komunikatów daje nam dynamiczny obraz struktur pokazując sposób współpracy powiązanych obiektów.

2.5 Zastosowanie struktur obiektowych do stworzenia systemu analizy VFEM++

2.5.1 Struktura ogólna

Ze względu na czytelność struktury obiektowe programu VFEM zostaną przedstawione na kilku wzajemnie uzupełniających się schematach. Na Rys. 2.8 pokazana jest zasadnicza struktura powiązań obiektu **Konstrukcja**. Obiekt ten jest głównym obiektem programu opisującym analizowaną konstrukcję. Jest rodzajem bazy danych przechowującym wszystkie dane składające się na model konstrukcji. **Konstrukcja** ponadto współpracuje

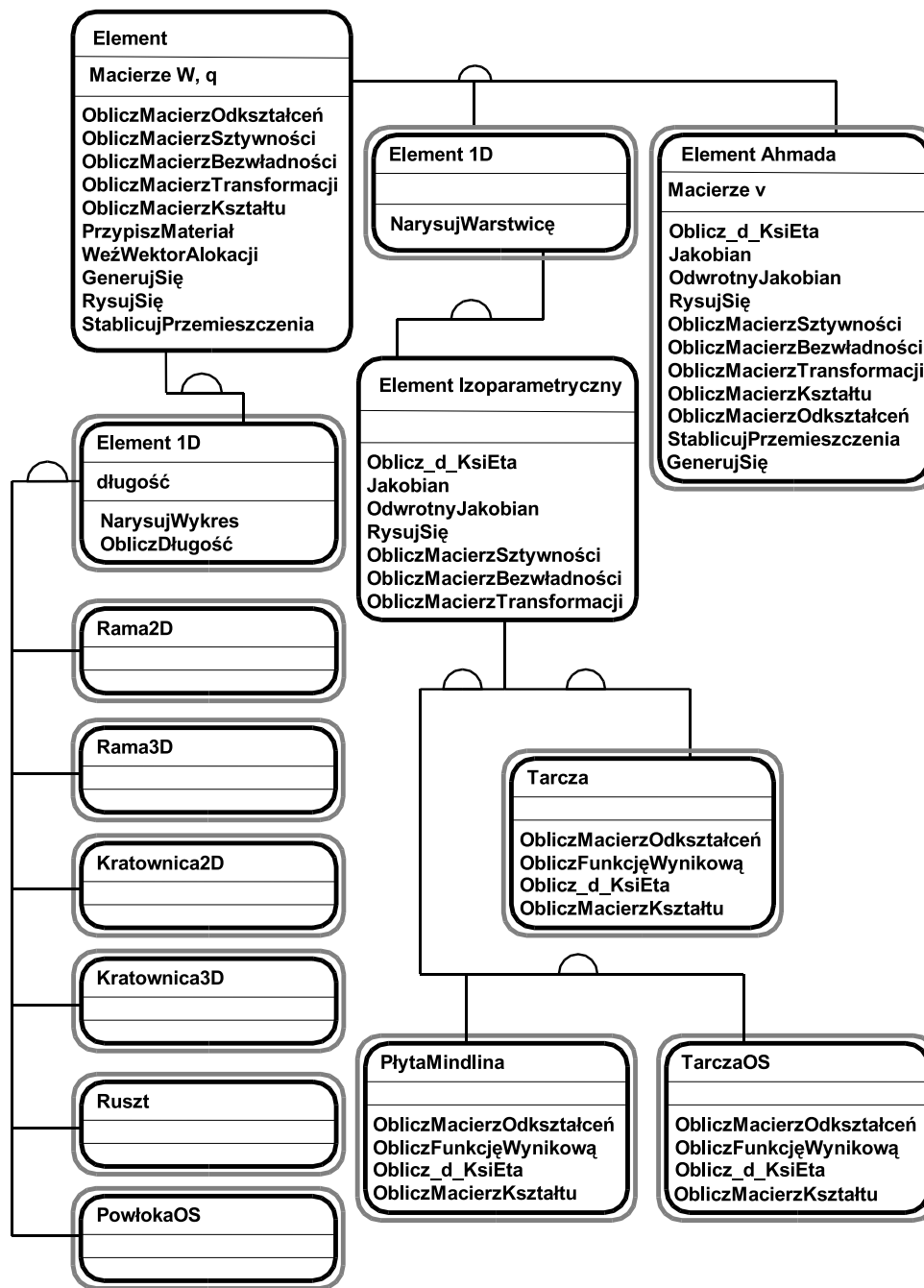


Rysunek 2.8: Ogólna struktura systemu

z innymi obiektami, jak na przykład: `EkranGraficzny` reprezentującego wyjście graficzne, na którym jest rysowana konstrukcja (np. okno Windows, drukarkę, czy plik graficzny). Poszczególne części konstrukcji opisywane są przez obiekty tworzące wraz z obiektem `Konstrukcja` strukturę całość-część. Zdziwienie może budzić fakt, że obiekty takie, jak: `Material`, czy `ObciążenieElementowe` są częściami konstrukcji. Wydawałoby się, że powinny być częściami elementu. Zrobiono tak dlatego, żeby nie tworzyć wielu identycznych obiektów. Gdyby `Element` był tylko częścią obiektu `Element`, a konstrukcja zawierałaby np. 1000 elementów o tym samym materiale, wówczas trzeba by utworzyć 1000 egzemplarzy obiektów klasy `Material` o identycznych parametrach. W przypadku, gdy `Material` jest też częścią obiektu `Konstrukcja`, tworzony jest pojedynczy egzemplarz obiektu `Material` i może być przypisany jako część wielu elementom. Podobna zasada dotyczy obiektów `ObciążenieElementowe`, `ZwolnienieElem` (obiekt ten reprezentuje wolny stopień swobody elementu, tj. taki, który nie jest "podłączony" do żadnego innego elementu czy węzła np. przegub).

2.5.2 Klasy elementów skończonych

Hierarchia klasy `Element` jest reprezentacją obiektową poszczególnych rodzajów elementów skończonych, w jakie wyposażony jest program VFEM. Schemat hierarchii pokazany jest na Rys. 2.9. Na rysunku tym pominięto powiązania klasy `Element` z innymi niż elementy skończone klasami. Abstrakcyjna bazowa klasa `Element` zawiera metody, które musi zawierać każdy element skończony. Klasa ta zawiera też dwa atrybuty. Jednym jest macierz q zawierająca wektor uogólnionych przemieszczeń węzłowych (węzłów, na



Rysunek 2.9: Hierarchia klas elementów skończonych

których rozpięty jest element) w układzie lokalnym elementu. Wektor taki dostarczany jest każdemu elementowi przez rozwiązujący zadanie obiekt klasy *AnalizaMES* po rozwiązaniu zadania, umożliwiając elementowi obliczenie wyników. Struktury powiązań klasy *AnalizaMES* zostaną zaprezentowane w dalszej części pracy. Macierz W jest wektorem sił wewnętrznych. Taka konstrukcja klasy nie narzuca sposobu numerycznej realizacji poszczególnych funkcji elementu. Poszczególne macierze mogą być obliczane przez numeryczne całkowanie albo zdefiniowane analitycznie. Klasa *Element* nie jest jedyną abstrakcyjną klasą w hierarchii. W klasyfikacji elementów wymieniono bardziej szczegółowe klasy

elementów, jak: elementy jednowymiarowe (klasa `TElement1D`), elementy dwuwymiarowe (klasa `TElement2D`), a w klasie elementów dwuwymiarowych-elementy izoparametryczne reprezentowane przez `ElementIzoparametryczny`. Dopiero na bazie tych klas buduje się klasy-i-objekty opisujące konkretne elementy skończone. Klasa `Element1D` posiada jeden atrybut, jakim jest *długość* elementu. Dziedziczą z niej klasy reprezentujące jednowymiarowe elementy (`RamaPłaska` itp.). W tych klasach-i-objektach zdefiniowano wszystkie potrzebne do obliczeń funkcje zadeklarowane w klasie `Element` i `Element1D`, których nie umieszczono na schematach klas elementów jednowymiarowych na Rys. 2.9 ze względu na brak miejsca. Zastosowany mechanizm funkcji polimorficznych pozwala na definicję tej samej funkcji w każdej z klas-i-objektów odrębnie. W elementach prętowych zastosowano analityczną definicję funkcji obliczających macierze sztywności, bezwładności i macierzy wpływu podłoża Winklera. W Elementcie powłoki obrotowo-symetrycznej macierze te obliczane są w sposób numeryczny, z zastosowaniem kwadratury Gaussa w 7 punktach (w/g zaleceń zawartych w [50]).

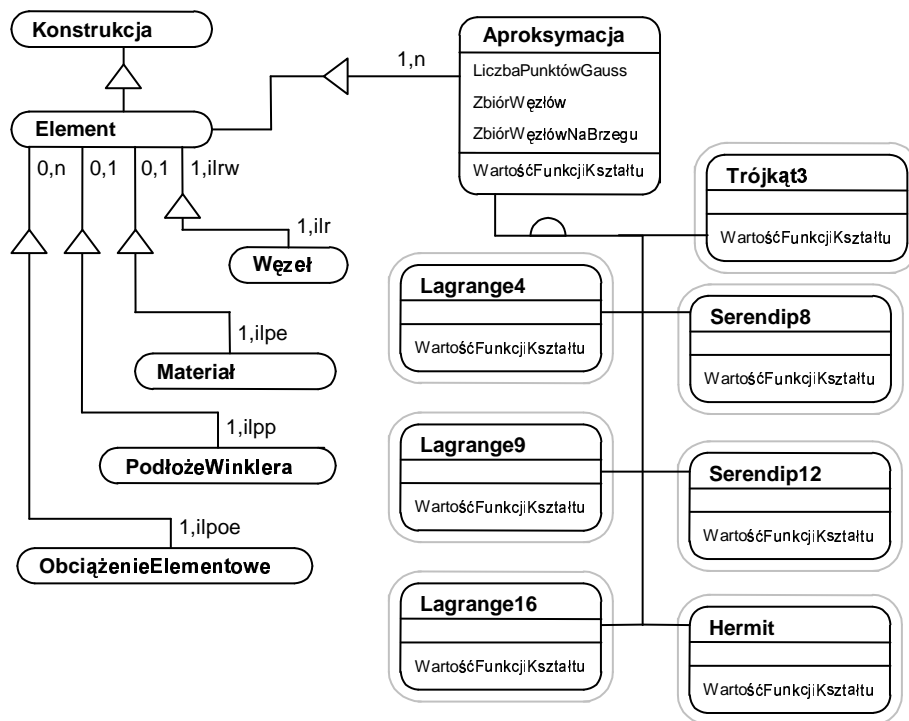
W klasie elementów dwuwymiarowych zastosowano w systemie jedynie elementy izoparametryczne. Gdyby zaistniała potrzeba utworzenia nieizoparametrycznego elementu płaskiego, dziedziczyłby on z klasy `Element2D`. Podejście izoparametryczne jest na tyle ogólne, że pozwala zdefiniować funkcje obliczające macierze elementu już w abstrakcyjnej klasie `IzoparametrycznyElement2D`. Jediną różnicą pomiędzy różnymi typami elementów izoparametrycznych w funkcjach obliczających macierze jest różniczkowy element objętości:

$$K^e = \int_V B^T D B dV. \quad (2.1)$$

Dla tarczy $dV = h \cdot dx \cdot dy$, zaś dla tarczy w stanie obrotowej symetrii mamy $dV = 2 \cdot \pi \cdot r \cdot dr \cdot dz$. Funkcja `D_KsiEta` zwraca właśnie tę wartość. Widać na Rys. 2.9, jak łatwo wprowadzić do systemu nowy element bazujący na koncepcji izoparametrycznej. Wystarczy zdefiniować tylko funkcję liczącą macierz odkształceń, macierz kształtu (tu w odróżnieniu od elementów jednowymiarowych macierz kształtu jest funkcją dwóch zmiennych), `D_KsiEta` oraz `FunkcjęWynikową` (ta ostatnia służy do obliczania wszystkich funkcji wynikowych, jakie przewidujemy dla danego typu elementu). Zupełnie innym elementem jest uwidoczniiony na schemacie powłokowy element `ElementAhmada`. Dlatego konieczne było zdefiniowanie w tej klasie odrębnych wersji większości jego funkcji. Klasa ta posiada ponadto atrybut `v`. Są to wektory normalne do powierzchni środkowej elementu w jego węzłach służące do opisu geometrii elementu. Szczegółowy opis tego elementu można znaleźć w pracy [50].

2.5.3 Aproksymacja i geometria elementów skończonych

Klasa elementu izoparametrycznego zdefiniowana jest tak, żeby wszystkie informacje dotyczące aproksymacji i geometrii węzłów pobierane były z obiektu `Aproksymacja`. Jest to pokazane na Rys. 2.10. `Aproksymacja` jest częścią obiektu `Element`. W tym obiekcie zawarte są: informacje dotyczące geometrii węzłów, węzły znajdujące się na brzegu elementu (potrzebnych do narysowania elementu), liczba punktów Gaussa potrzebna do całkowania, funkcje kształtu i ich pochodne do drugiego rzędu włącznie. Obiekty `Element` i `Aproksymacja` są tworzone niezależnie i współpracują ze sobą. Dzięki takiemu rozwiązaniu wystarczy stworzyć tylko jedną nową klasę elementu izoparametrycznego. Obiekty tej

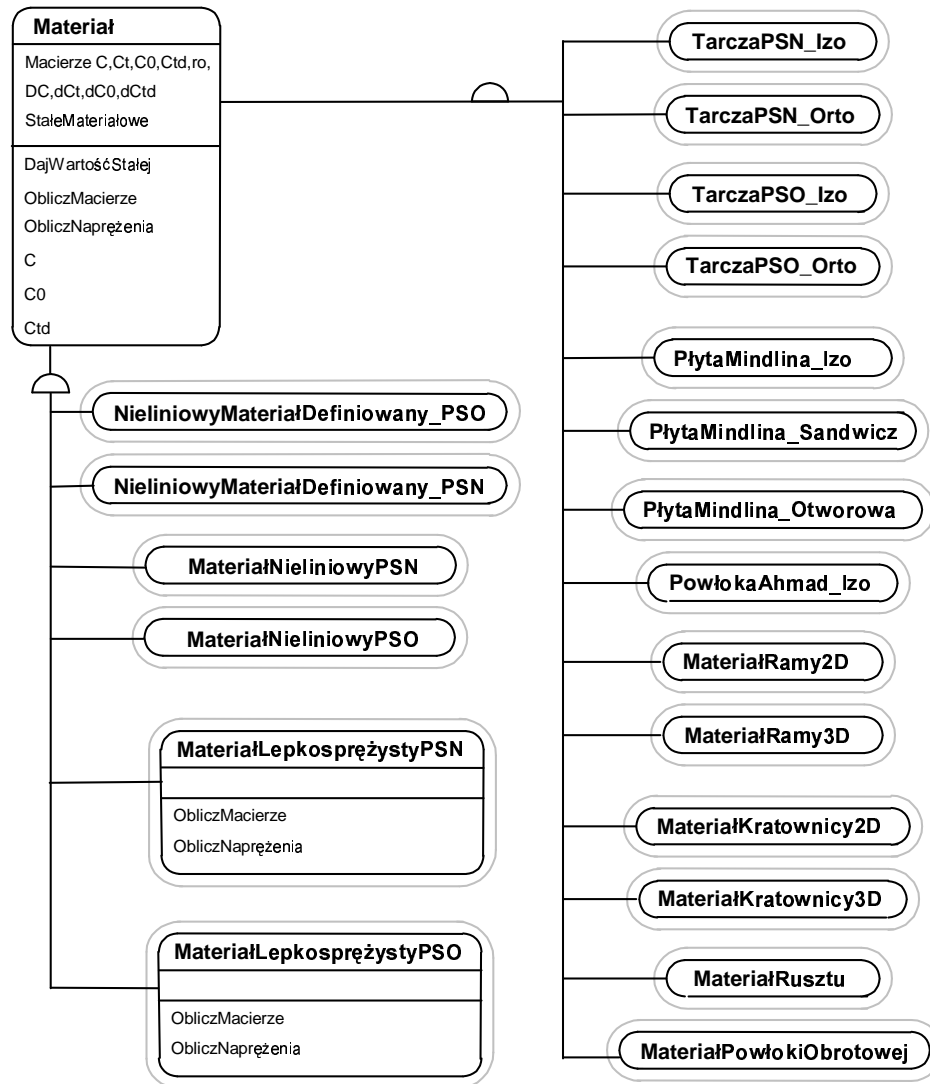


Rysunek 2.10: Obiekt Element i jego części. Hiererchia obiektu aproksymacja

klasy będą automatycznie działać z wszystkimi już istniejącymi klasami aproksymacji. Tak był wprowadzany do programu VFEM element Powłoka Ahmada. Przed wprowadzeniem tego elementu były już w programie aproksymacje przedstawione na Rys. 2.10, współpracujące z elementami tarczy i płyty (tj. wszystkie oprócz aproksymacji Hermita). Wystarczyło stworzyć klasę PowłokaAhmada. Obiekty tej klasy zaczęły współpracować z klasami aproksymacji istniejącymi już w programie. Taka konstrukcja znacznie ułatwia wprowadzanie, zarówno nowych klas elementów, jak i nowych klas aproksymacji. Nowo dodawany element będzie współpracował z istniejącymi klasami aproksymacji. Nowa aproksymacja wprowadzana do programu będzie współpracować z wszystkimi typami elementów. Oczywiście nie każda kombinacja element-aproksymacja ma sens np. tarcza nie może współpracować z aproksymacją Hermita. Najważniejsze jest jednak to, aby nie było żadnych przeszkód formalnych, czy utrudnień w zastosowaniu dowolnej kombinacji element - aproksymacja. Ocena sensowności połączenia może być wykonana na etapie wprowadzania danych. Tak się dzieje podczas wprowadzania danych w systemie VFEM++, a ewentualne niezgodności są sygnalizowane błędem.

2.5.4 Klasy materiałów

Zadaniem obiektów klasy **Materiał** jest reprezentowanie poszczególnych typów materiałów. Obiekty te przechowują stałe materiałowe oraz posiadają funkcje budujące macierze materiałowe. Dla elementów prętowych, gdzie macierze (sztywności, bezwładności itp.) obliczane są analitycznie, jedyną funkcją obiektu **Materiał** jest przechowywanie stałych



Rysunek 2.11: Hierarchia klas materiałowych

materiałowych. W przypadku materiałów nieliniowych w klasie **Materiał** definiowane są funkcje materiałowe. Jako stałe materiałowe rozumie się tu także charakterystyki przekrojowe takie jak pole przekroju, czy moment bezwładności. Zasadniczo powinny być stworzone klasy o nazwie np. **Geometria** opisujące własności przekrojów. Takie podejście byłoby uzasadnione w przypadku bardziej inżynierskiego podejścia dającego możliwość definicji złożonych przekrojów, np. kształtowników walcowanych, czy innych złożonych przekrojów. Dla uproszczenia zrezygnowano tu jednak z tak detalicznego podejścia.

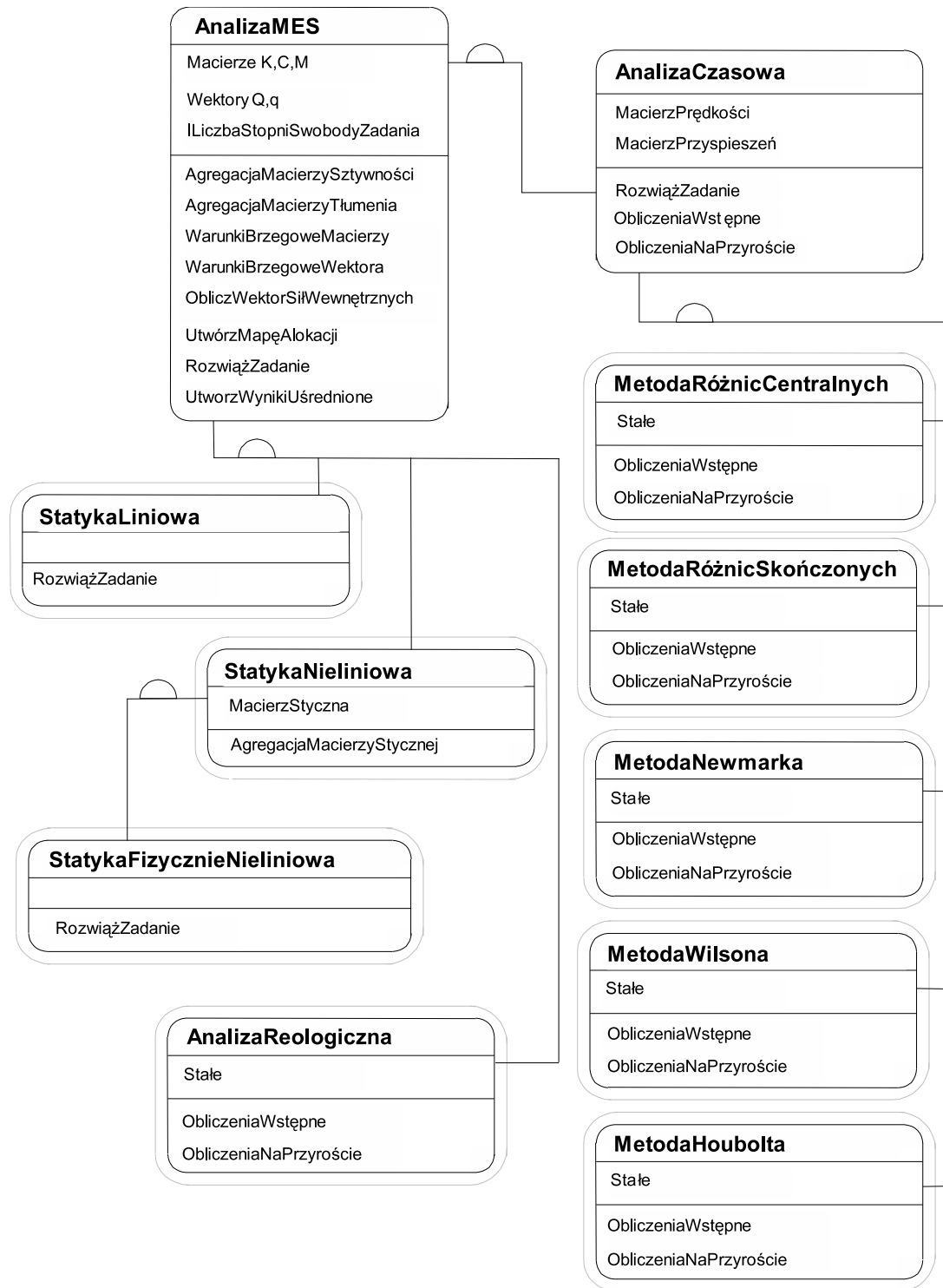
Każdy z typów elementów skończonych posiada co najmniej jeden odpowiadający mu materiał. Elementy prętowe zdefiniowane analitycznie nie mogą posiadać innych materiałów, jak tylko ten, dla którego wyprowadzono analityczne wzory na macierz sztywności i inne macierze. Podejście numeryczne, gdzie do obliczeń potrzebna jest macierz materiałowa, umożliwia zdefiniowanie wielu różnych typów materiałów dla jednego typu elementu skończonego. Niektóre elementy skończone istniejące w systemie posiadają kilka

typów materiałów, do wyboru przez użytkownika. Są to elementy tarczowe posiadające materiał izotropowy oraz materiał warstwowo-izotropowy. Element płyty Mindlina ma do dyspozycji trzy materiały: Izotropowy, Sandwiczowy oraz materiał płyty z wyciętymi prostokątnymi otworami. Hierarchię klas **Materiał** obrazuje Rys. 2.11. Z klasy **Materiał** dziedziczą bezpośrednio klasy-i-objekty reprezentujące poszczególne materiały liniowe. W abstrakcyjnej klasie **Materiał** zdefiniowano wszystkie funkcje potrzebne do działania obiektu tej klasy. W szczególności zadeklarowano w tej klasie dwie funkcje abstrakcyjne. Dziedziczące z klasy bazowej materiały muszą te funkcje zdefiniować. Pierwszą z nich jest funkcja **ObliczMacierze**, która ma za zadanie obliczyć wszystkie macierze materiałowe używane przez dany materiał, wraz z ich pochodnymi względem stałych materiałowych potrzebnymi w analizie wrażliwości parametrycznej. Drugą funkcją to funkcja **ObliczNapężenia**, reprezentująca prawo konstytutywne dla danego materiału. Wszystkie używane macierze materiałowe jak i funkcje umożliwiające dostęp do nich, zadeklarowane są w bazowej klasie abstrakcyjnej **Materiał**. Może się to wydawać pozornie nielogiczne, ponieważ w rezultacie każdy materiał posiada wszystkie funkcje. Po co na przykład materiałom dla zagadnienia statyki liniowej funkcje materiałowe, zwracające macierze C_{td} czy C_0 , właściwe dla materiałów lepkosprężystych? Może się jednak zdarzyć, że będziemy chcieli, aby część konstrukcji poddanej analizie reologicznej miała właściwości wyłącznie sprężyste. Innymi słowy materiał sprężysty musi się dać zastosować nie tylko w statyce, czy dynamice liniowej, ale również w innych typach analiz. Teoretycznie hierarchia klas nie powinna stwarzać formalnych przeszkód stosowania dowolnych materiałów w dowolnych typach analiz. Dlatego w klasie bazowej zdefiniowano wszystkie znane typy macierzy materiałowych oraz funkcji materiałowych.

2.5.5 Obiektowy model analizy skończenie elementowej

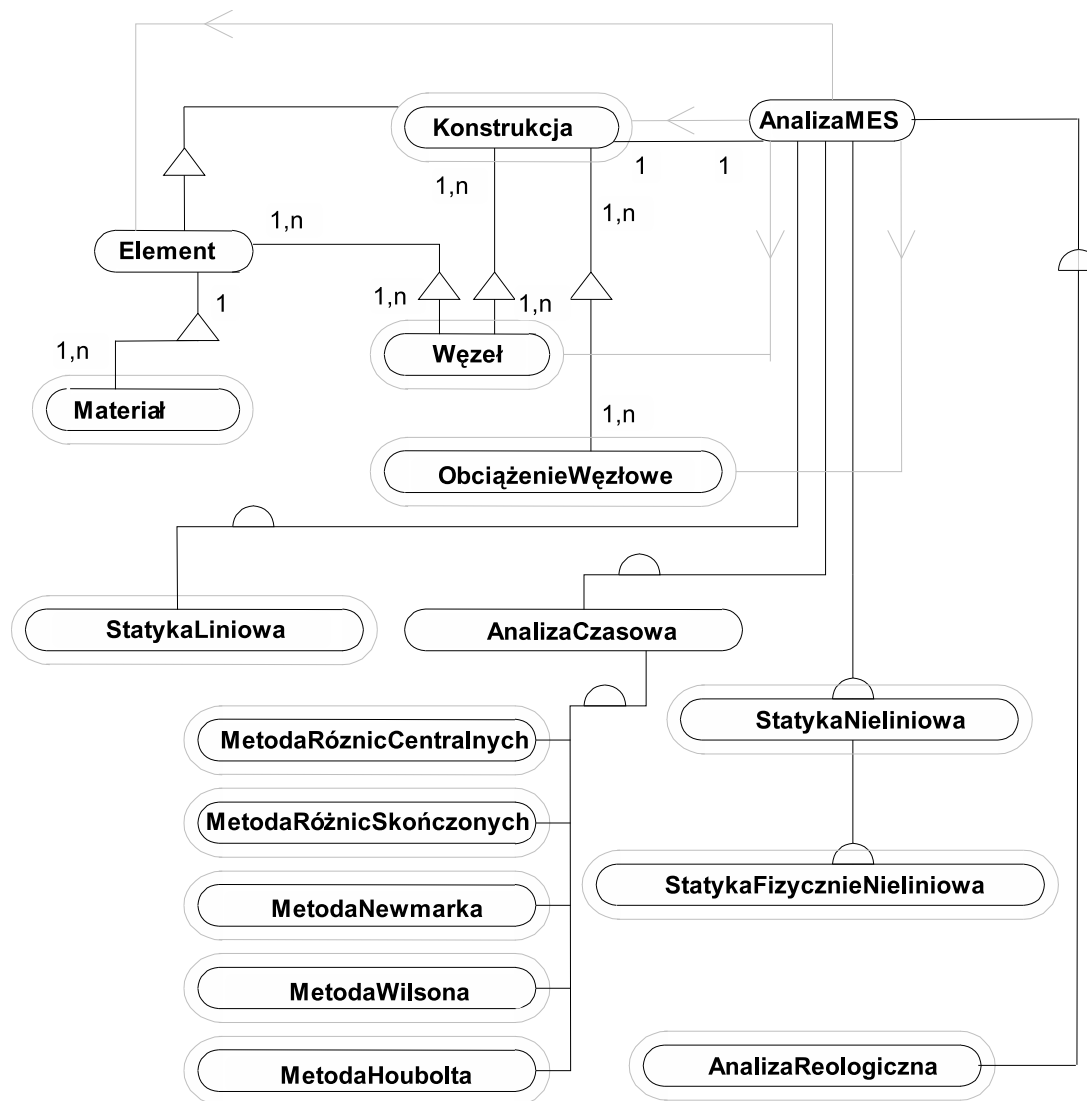
Obiekty klasy **AnalizaMES** służą do wykonania analizy skończenie elementowej. Schemat hierarchii klasy **AnalizaMES** zawiera Rys. 2.12. W bazowej klasie deklarowane są wspólne dla wszystkich typów analiz atrybuty i metody. Są to globalne macierze sztywności, bezwładności, macierz tłumienia, wektory obciążeń i przemieszczeń. Zasadniczą funkcją jest funkcja **RozwiążZadanie**. W ciele tej funkcji znajduje się kod algorytmu realizującego dany rodzaj analizy. Bezpośrednio z klasy **AnalizaMES** wyprowadzona jest klasa **StatykaLiniowa**. Dziedzicząca również z klasy **AnalizaMES**, klasa **StatykaNieliniowa** realizuje analizę fizycznie nieliniową. Ten rodzaj analizy dostępny jest tylko dla kratownic płaskich i przestrzennych, gdyż tylko te obiekty mają zdefiniowane procedury obliczające macierz styczną. Łatwo jednak rozszerzyć możliwości systemu o analizę nieliniową innych konstrukcji definiując funkcję obliczającą macierz styczną dla innych elementów. Z pewnością takie funkcje zostaną do systemu dodane w niedalekiej przyszłości.

Dziedzicząca z klasy **StatykaNieliniowa** klasa **StatykaFizycznieNieliniowa** reprezentuje, zgodnie z nazwą, statykę fizycznie nieliniową i w aktualnej wersji programu jest dostępna tylko dla tarcz w płaskim stanie odkształcenia. Tylko dla tarcz PSO istnieją klasy nieliniowego materiału. W razie pojawienia się konieczności analizowania innych konstrukcji z nieliniowościami materiałowymi można łatwo dodać stosowne klasy materiałów dla tych konstrukcji. Klasa **AnalizyCzasowa** łączy w sobie wszystkie wspólne cechy metod bezpośredniego całkowania równania ruchu. Każda z metod bezpośredniego całkowania równania ruchu składa się z obliczeń wstępnych oraz powtarzanych w pętli obliczeń na przyszłość. Tylko te dwie funkcje wystarczy zdefiniować w każdej z klas dziedziczących,



Rysunek 2.12: Schemat hierarchii klas analizy skończenie elementowej

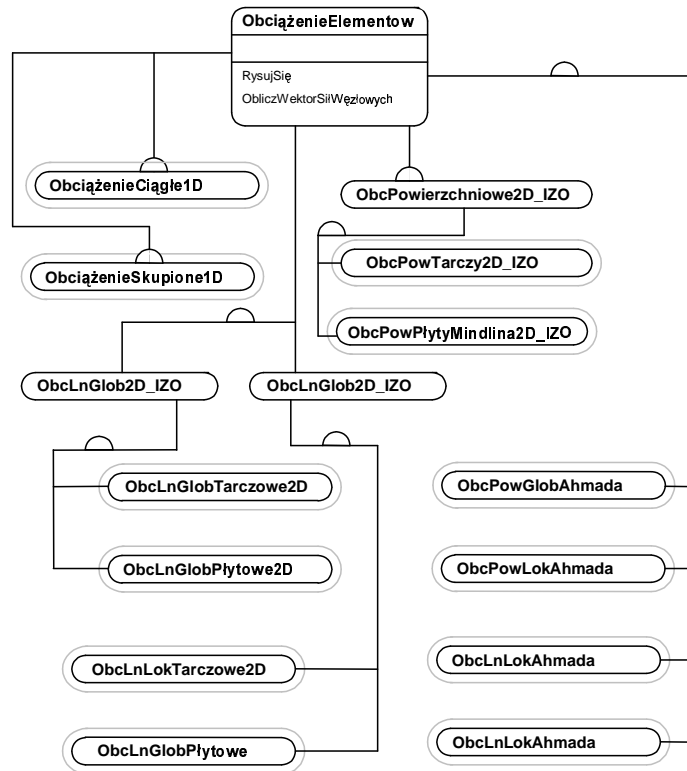
reprezentujących konkretną metodę całkowania równania ruchu. Na Rys. 2.13 pokazano schemat powiązań klasy *AnalizaMES*. Widać tu, że podczas analizy solver wysyła komunikaty do czterech innych obiektów: *Konstrukcji*, *Elementu*, *Węzła* oraz *ObciążeniaWęzłowego*.



Rysunek 2.13: Schemat powiązań klasy analizy skończenie elementowej

2.5.6 Klasy obciążeń elementowych

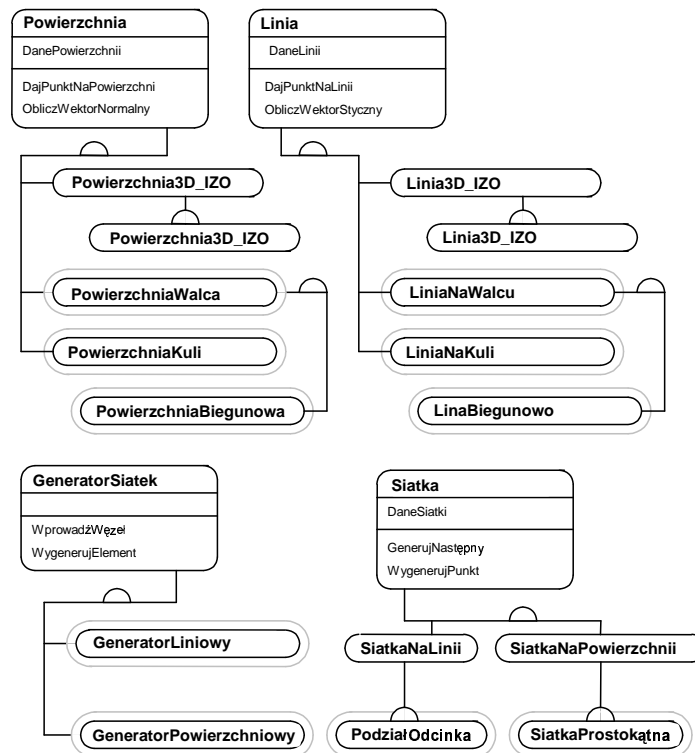
Każdy z poszczególnych typów elementów skończonych może posiadać różne rodzaje obciążeń elementowych. Struktura *gen-spec* jest więc i w tym przypadku bardzo pomocna. Podstawową bazową klasą jest klasa `ObciążenieElementowe`. Wyprowadzone z niej są abstrakcyjne klasy dla poszczególnych typów elementów, a następnie dla poszczególnych typów obciążeń. Jediną funkcją, jaką trzeba zdefiniować w każdej z klas, jest funkcja `ObliczWektorSiłWęzłowych` służąca zgodnie z nazwą do obliczenia wektora obciążeń węzłowych elementu dla danego obciążenia międzywęzłowego. Hierarchia klas pokazana jest na Rys. 2.14. Ze względu na długość nazw dokonano w nich skrótów. *Pow* oznacza obciążenie powierzchniowe, *Ln* - obciążenie działające wzdłuż linii. *Glob* - obciążenie globalne (obciążenie, którego składowe podaje się w kartezjańskim układzie globalnym). *Lok* - to obciążenie, które działa stycznie do powierzchni lub linii działania.



Rysunek 2.14: Hierarchia klas obciążeń elementowych

2.5.7 Klasy obiektów geometrycznych i generatora siatek regularnych

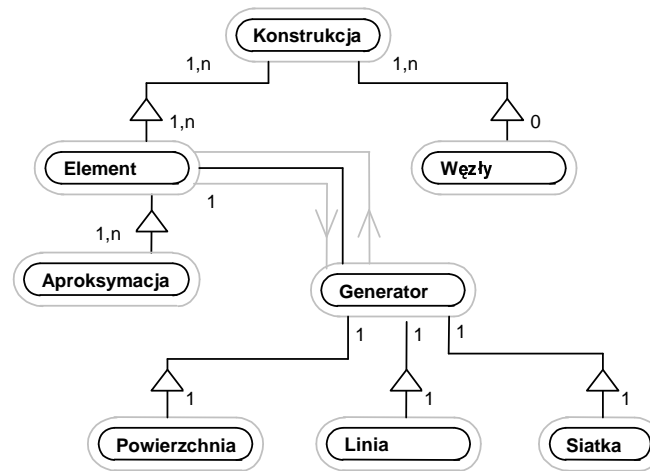
Klasy realizujące generację siatki powinny być jak najbardziej uniwersalne. Zagadnienie generacji siatek opisują obiekty należące do trzech różnych hierarchii. Pierwsza z nich to klasy obiektów geometrycznych, na których będzie generowana siatka. Mogą to być powierzchnie lub linie, w zależności od jakiego typu elementy będą generowane. I tak mamy powierzchnię rozpiętą na czterech (Lagrange4) lub ośmiu (Serendip8) punktach. Jest to klasa `Powierzchnia3D_IZO` lub w przypadku zadań płaskich `Powierzchnia2D_IZO`. Możemy generować siatki na powierzchni zdefiniowanej j/w , ale we współrzędnych walcowych lub kulistych obsługują to klasy `PowierzchniaWalca` i `PowierzchniaKuli`. Klasy te posiadają dwie funkcje składowe: jedna oblicza współrzędne kartezjańskie punktu o zadanych współrzędnych naturalnych powierzchni, czy linii, druga zaś wyznacza wektor normalny (w przypadku klasy `Powierzchnia`), albo wektor styczny (w przypadku klasy `Linia`). W zadaniach płaskich powierzchnię możemy zdefiniować we współrzędnych biegunowych przy pomocy obiektów klasy `PowierzchniaBiegunowa`. Takie same możliwości mamy w przypadku generacji elementów jednowymiarowych na zadanej linii. Drugim typem klas współpracującym przy generacji siatek jest klasa `Siatka`. Klasy typu siatka opisują rodzaj generowanej siatki. Dostępne są dwie siatki: jedna to siatka prostokątna o zadanej liczbie elementów skończonych na bokach prostokąta oraz dla siatek liniowych - siatka dzieląca zadany odcinek na ustaloną liczbę części. Metoda `Wygeneruj punkt` zawarta w



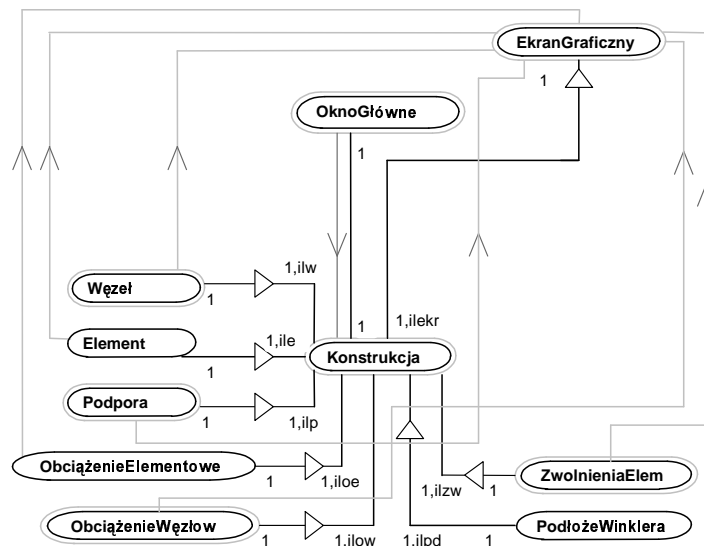
Rysunek 2.15: Hierarchie klas: Powierzchni, Linii, Generatora, Siatki

klasie **Siatka** ma za zadanie przeliczyć współrzędne węzła ze współrzędnych naturalnych elementu skończonego do współrzędnych parametryzujących powierzchnię. Druga z funkcji (**Następny**) decyduje o tym, kiedy zakończyć proces generacji zwracając wtedy wartość logiczną *false*. Proces generowania siatek nadzorowany jest przez obiekty klasy **Generator**. Są dwie klasy-i-obiekty tego typu: generator dla elementów powierzchniowych oraz generator dla elementów liniowych.

Na Rys. 2.16 przedstawiono strukturę powiązań obiektu **Generator** z innymi obiektami współpracującymi podczas generacji siatek regularnych. Przez siatkę regularną rozumie się taką siatkę, której liczba elementów skończonych w wygenerowanym modelu jest znana przed rozpoczęciem generacji. Zastosowania innych struktur wymaga generator siatek nieregularnych np. triangularyzacja Delaunay'a. Ten generator będzie szczegółowo opisany w dalszej części pracy. Dla siatek regularnych proces generacji przebiega następująco: **Generator** dostaje obiekt reprezentujący element skończony. Zadaniem generatora jest ustalić listę węzłów, na których element jest rozpięty. W tym celu **Generator** wywołuje funkcję elementu **GenerujSię** (patrz Rys. 2.9). W funkcji tej **Element** pobiera z obiektu **Aproksymacja** współrzędne naturalne węzłów i kolejno zwraca je **Generatorowi**, który z kolei wywołuje składową obiektu **Siatka** przeliczającą współrzędne naturalne elementu na współrzędne naturalne generowanego obszaru. Te z kolei przeliczane są przez obiekt klasy **Powierzchnia** albo **Linia** na współrzędne kartezjańskie. Węzeł o takich współrzędnych wprowadzany jest do bazy węzłów. Cykl ten jest powtarzany dla wszystkich elementów. Taka konstrukcja generatora daje bardzo dużą uniwersalność. Generator nie musi mieć żadnych



Rysunek 2.16: Struktura powiązań obiektu generator



Rysunek 2.17: Struktura powiązań z obiektem EkranGraficzny oraz komunikacja z nim

danych o elementach i może generować elementy dowolnego typu. Wystarczy, aby każdy typ elementu posiadał funkcję polimorficzną `GenerujSię`.

2.5.8 Graficzny interfejs użytkownika

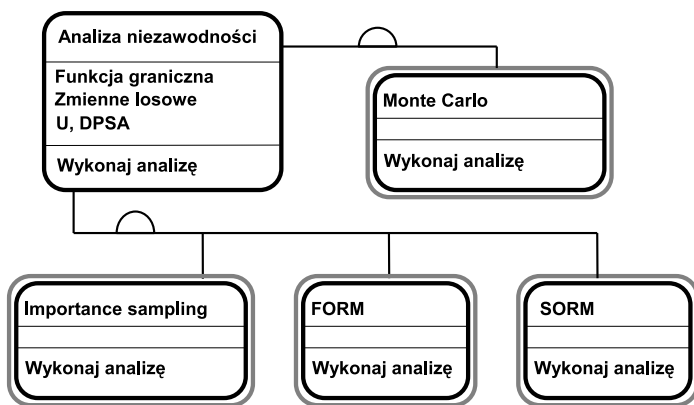
Klasą reprezentującą okno do wyświetlania schematu konstrukcji jest `EkranGraficzny`. Zawiera on wszystkie funkcje potrzebne do operacji graficznych: rysowania, drukowania i zapisu schematu konstrukcji. Jakikolwiek inny obiekt wykorzystujący funkcje graficzne systemu musi mieć powiązania z tym obiektem. W ekranie graficznym zdefiniowane są funkcje realizujące przekształcenie aksonometryczne oraz rzuty schematu. Atrybutami ekranu graficznego są wszystkie parametry wyświetlania. Liczba egzemplarzy obiektu w

programie odpowiada liczbie otwartych przez użytkownika okien graficznych, lub liczbie plików graficznych jakie będą zapisane w danym zadaniu. Dzięki temu jest możliwość utworzenia kilku różnych widoków tej samej konstrukcji równocześnie.

Pokazana na Rys. 2.17 struktura powiązań oraz sposób obsługi grafiki nie są zbyt skomplikowane. Program VFEM++ nie posiada wprowadzania danych o konstrukcji przy pomocy edytora graficznego. Przy tej metodzie wprowadzania danych struktury graficzne byłyby bardziej skomplikowane.

2.5.9 Klasy analizy niezawodnościowej

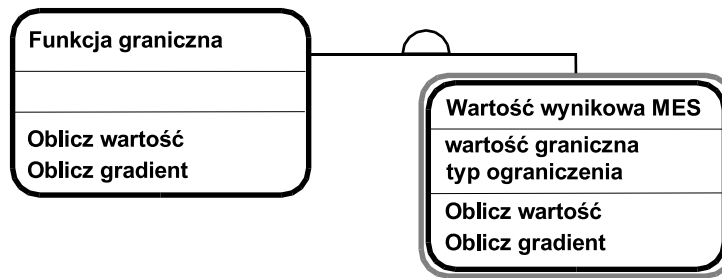
Wobec dużej różnorodności metod analizy niezawodnościowej podejście obiektowe jest również w tym przypadku bardzo efektywnym narzędziem programowania takich obliczeń. Istnieją bowiem nie tylko najrozmaitsze metody analizy niezawodności takie jak FORM (First Order Reliability Method), SORM (Second Order Reliability Method), Monte Carlo czy Importance Sampling, ale i również wiele alternatywnych procedur składających się na te metody. Można bowiem w ramach metod FORM czy SORM stosować różne algo-



Rysunek 2.18: Hierarchia klas analizy niezawodnościowej

rytmy znajdowania punktu projektowego. Można stosować różne metody transformacji zmiennych losowych do przestrzeni standardowej (np. Natafa czy Rossenblatta). Istnieje też wiele różnych parametrów konstrukcji, których losowość życzymy sobie uwzględnić w naszej analizie, takich, jak: stałe materiałowe, parametry kształtu konstrukcji, czy obciążenie. Zaprogramowanie tych wszystkich elementów analizy niezawodnościowej polegać będzie na stworzeniu odpowiednich struktur obiektowych zaprezentowanych poniżej.

Na Rys. 2.18 przedstawiono hierarchię klas analiz niezawodnościowych. W klasie bazowej zadeklarowano jedną abstrakcyjną funkcję `WykonajAnalizę` która jest definiowana w dziedziczących z niej konkretnych klasach reprezentujących poszczególne typy analiz. W będącym częścią tej pracy systemie zaimplementowano trzy typy analizy niezawodnościowej: Monte Carlo, FORM, SORM i Importance Sampling. Bazowa klasa posiada również kilka atrybutów wspólnych dla wszystkich klas bazowych takich jak, wskaźnik do funkcji celu, z którym ten obiekt współpracuje oraz zbiór zmiennych losowych, których wartości są potrzebne podczas analizy. Pewnego komentarza wymaga fakt, iż Analiza skończenie elementowa nie posiada wspólnej klasy bazowej z analizą niezawodnościową i opisaną poniżej analizą optymalizacyjną. Łatwo jednak zauważyć, że tych analiz nie ze

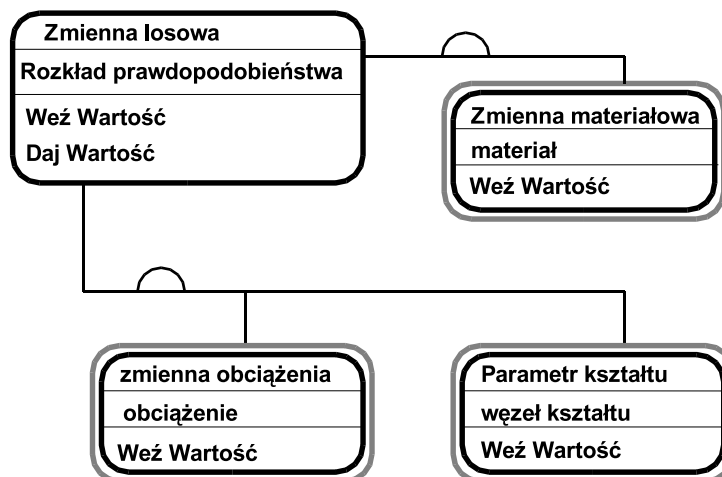


Rysunek 2.19: Hierarchia klas funkcji granicznej

sobą nie łączy. Poza funkcją realizującą analizę nie miałyby żadnych wspólnych cech dlatego uznano, że powinny one stanowić odrębne byty hierarchiczne. Istnieje natomiast współpraca pomiędzy analizą niezawodności a analizą MES Rys. 2.23, ponieważ to właśnie rezultaty obliczeń MES stanowią podstawę do obliczenia funkcji granicznej i jej gradientów. Ogólną postać funkcji granicznej, jaka istnieje w systemie, można przedstawić następująco:

$$R \leq R_l \text{ lub } R \geq R_l. \quad (2.2)$$

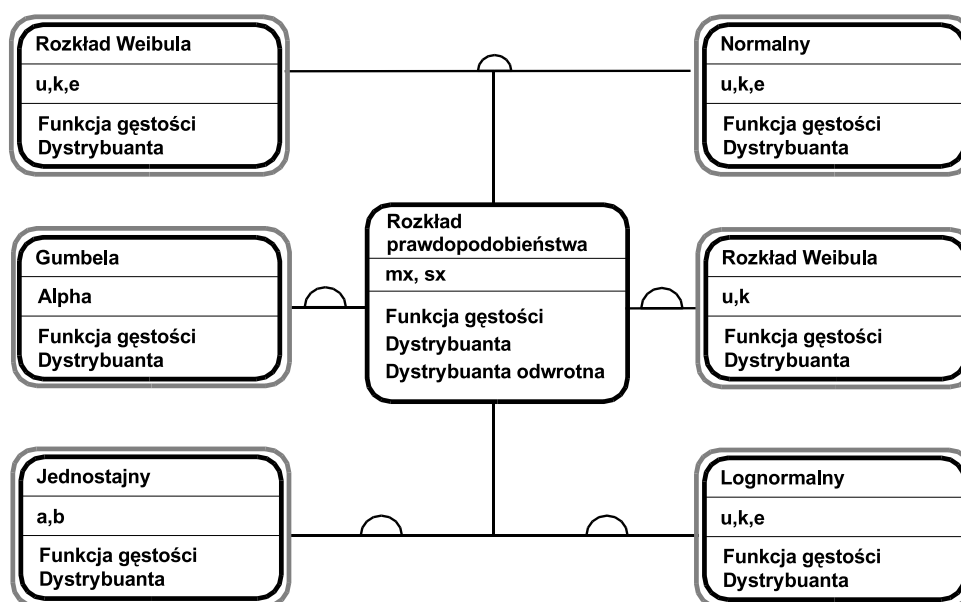
gdzie R stanowi dowolny wynik analizy skończenie elementowej taki, jak przemieszczenie węzłów czy naprężenie elementowe w dowolnym punkcie elementu, jak i ich wbudowane funkcje takie, jak wartości główne czy np. zredukowane naprężenia Hubera-Misesa. a R_l to wielkość graniczna, którą dana wielkość jest ograniczona od góry lub od dołu. Wyznaczaniem wielkości wynikowych MES zajmują się odrębne obiekty, stąd zaprezentowana na Rys. 2.19 hierarchia klas funkcji granicznej nie jest rozbudowana. Planuje się w przyszłości rozbudowę możliwości systemu o bardziej skomplikowane funkcje celu takie, jak zadawane przez użytkownika funkcje wartości wynikowych.



Rysunek 2.20: Hierarchia klas zmiennych losowych

W systemie można skorzystać z trzech rodzajów zmiennych losowych (Rys. 2.20). Są to: zmienna losowa reprezentująca losowy charakter stałej materiałowej, zmienna losowa obciążenia oraz parametr kształtu. Wspólną cechą zmiennych losowych jest funkcja wirtualna `WeżWartość` służąca do nadania wartości zmiennej. Funkcja jest polimorficzna, gdyż nie da się uogólnić sposobu nadawania wartości zmiennych losowych i dlatego każda klasa musi mieć własną definicję tej funkcji. Wspólnym dla wszystkich klas atrybutem zadeklarowanym w klasie bazowej jest obiekt `RozkładPrawdopodobieństwa` reprezentujący, zgodnie z nazwą, rozkład prawdopodobieństwa danej zmiennej.

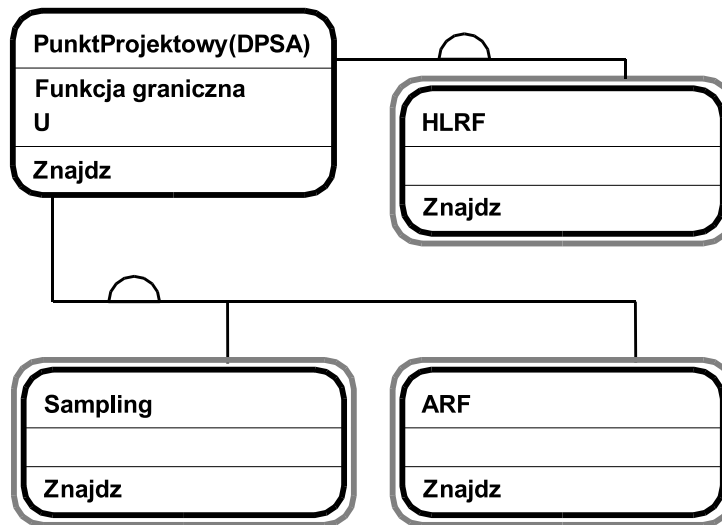
Rozkłady prawdopodobieństw również zostały zbudowane z użyciem struktur gen-spec. Wspólnymi atrybutami wszystkich rozkładów są: wartość średnia i odchylenie standardowe. W klasie bazowej ponadto postuluje się za pomocą deklaracji funkcji wirtualnych dwie cechy rozkładów prawdopodobieństw, a mianowicie: funkcję gęstości rozkładu oraz funkcję dystrybuanty rozkładu. Hierarchie klas oraz rodzaje klas rozkładów prawdopodobieństw pokazuje Rys. 2.21.



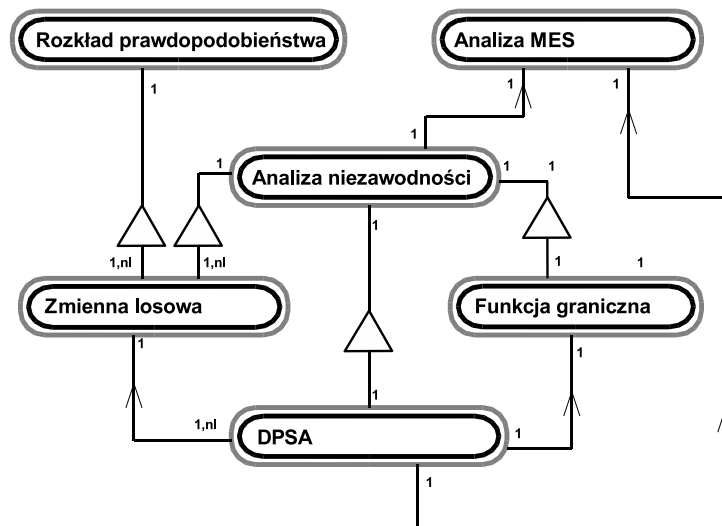
Rysunek 2.21: Hierarchia klas rozkładów prawdopodobieństwa

Do zaprogramowania algorytmów poszukiwania punktów projektowych została wykorzystana struktura gen-spec (Rys. 2.22). Pozwala to na zaimplementowanie wielu metod (takich jak `FORM` czy `ImportanceSampling`) na większym poziomie ogólności, co da możliwość łatwego wybierania dowolnej metody poszukiwania punktu projektowego. Wspólnymi atrybutami są obiekty funkcja celu i obiekt odpowiedzialny za transformację zmiennych losowych z przestrzeni oryginalnej do przestrzeni standardowej. Polimorficzną funkcją tej klasy jest funkcja realizująca algorytm poszukiwania punktu projektowego.

Struktury całość-część oraz sposób wymiany komunikatów między obiektami obrazuje diagram 2.23. Częścią zmiennej losowej jest rozkład prawdopodobieństwa. Zmienne losowe, wraz z funkcją graniczną i algorytmem poszukiwania punktu projektowego wchodzi w skład analizy niezawodnościowej. Do poszukiwania punktu projektowego potrzebne są wartości funkcji granicznej oraz dokonywanie zmian wartości zmiennych losowych, stąd też współpraca obiektu `AnalizaNieZawodności` z obiektami `FunkcjaGraniczna` i `ZmiennaLosowa`.



Rysunek 2.22: Hierarchia klas dla algorytmów poszukiwania punktów projektowych



Rysunek 2.23: Sieć wzajemnych powiązań obiektów analizy niezawodnościowej

Algorytm DPSA po zaktualizowaniu zmiennych losowych wywołuje obliczenia MES, stąd współpraca z obiektami analizy MES. Z analizą MES współpracują też niektóre obiekty analizy niezawodnościowej (Monte Carlo, i Importance Sampling).

2.5.10 Struktury klas optymalizacji

Można doszukać się pewnych podobieństw pomiędzy analizami niezawodności i optymalizacji. W optymalizacji istnieje bowiem funkcja zwana funkcją celu podobnie jak w niezawodności - funkcja graniczna. Zmienne projektowe optymalizacji są pewnym analogonem zmiennych losowych niezawodności. To formalne podobieństwo obu problemów nie uzasadnia jednak konieczności wyprowadzenia ich klas ze wspólnej klasy bazowej, gdyż mimo pozornych podobieństw są to zupełnie inne typy analiz. Dlatego też struktury

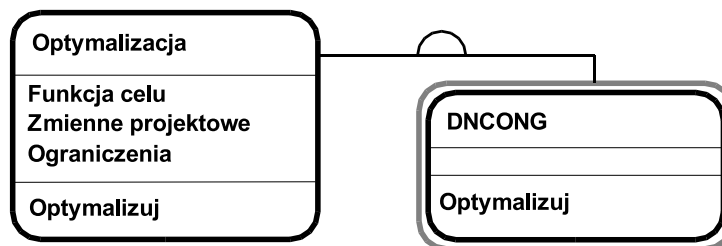
niezawodności i optymalizacji cechuje podobieństwo, ale są one odrębnymi strukturami danych. Hierarchię klas optymalizacji przedstawia Rys. 2.24.

W systemie istnieje jedna metoda optymalizacji, rozwiązująca problem nieliniowego programowania, oparta na algorytmie Schittkowskiego. W przyszłości planuje się rozbudowę systemu o inne procedury optymalizacyjne.

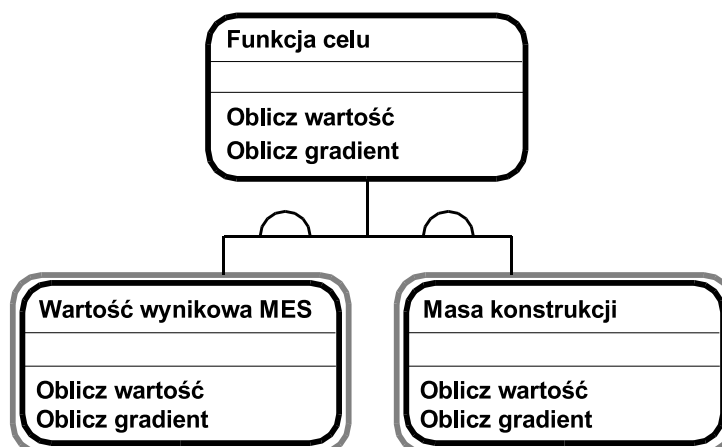
System zawiera dwa typy funkcji celu (Rys. 2.25) Są nimi funkcja celu pozwalająca na optymalizację wielkości wynikowych MES takich, jak przemieszczenia, naprężenia, oraz rozmaite funkcje (np wartości główne naprężeń czy odkształceń). Drugą funkcją celu jest masa konstrukcji. Ta funkcja celu pozwala rozwiązywać zagadnienia minimalizacji masy konstrukcji. Dwie funkcje wirtualne powinna definiować każda z klas-i-objektów funkcji celu. Jedna z nich wyznacza wartość funkcji celu, a druga jej gradient.

Rysunek 2.26 przedstawia reprezentację obiektową zmiennych projektowych. System zawiera dwie klasy zmiennych projektowych: klasę zmiennych materiałowych, gdzie zmienną projektową jest stała materiałowa oraz parametr kształtu, gdzie zmienną jest współrzędna węzła parametryzującego kształt konstrukcji. Współrzędne węzłów siatki MES zależne są od parametrów kształtu.

Bazowa klasa ograniczeń posiada jeden atrybut typu logicznego, którego wartość true oznacza, że dane ograniczenie jest ograniczeniem równościowym. Istnieje w systemie jedna klasa-i-objekt ograniczenia (Rys. 2.27). Jest to ograniczenie na wartość wynikową MES. Atrybutem tego ograniczenia jest pewna wielkość graniczna (podobnie, jak dla funkcji granicznych niezawodności (2.2). Dana wartość wynikowa może być mniejsza

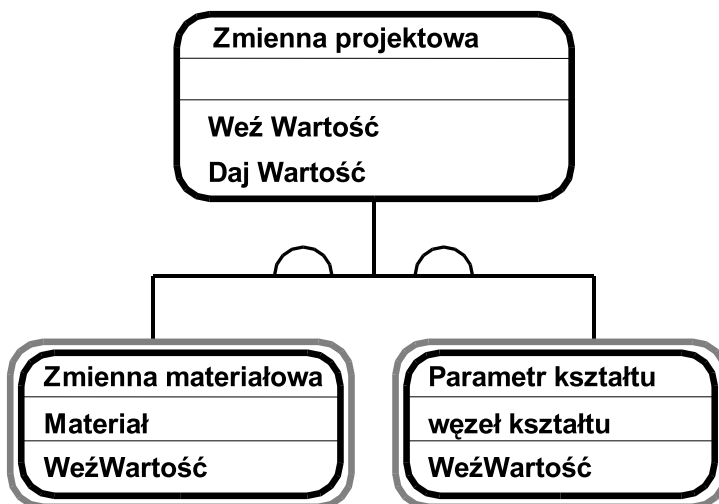


Rysunek 2.24: Hierarchia klas optymalizacji

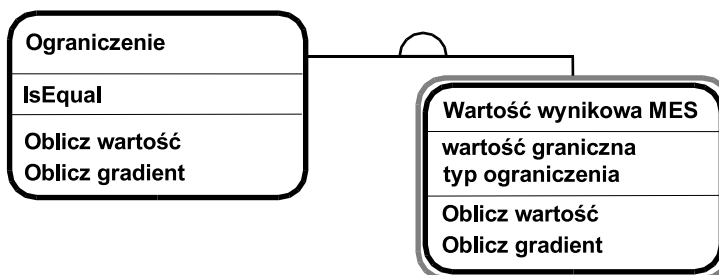


Rysunek 2.25: Hierarchia klas funkcji celu

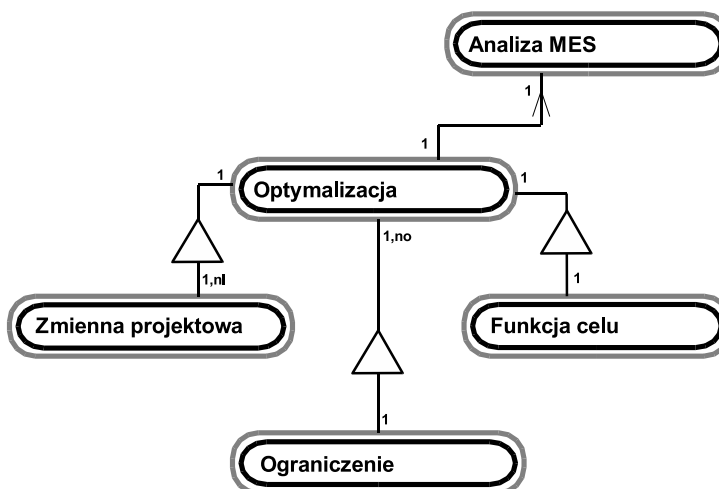
(lub większa, a w przypadku ograniczenia równościowego - równa) od założonej przez użytkownika wielkości granicznej. Wirtualnymi funkcjami zadeklarowanymi w klasie bazowej są funkcje liczące wartość i gradient ograniczenia. Rysunek 2.28 zawiera schemat



Rysunek 2.26: Hierarchia klas zmiennych projektowych



Rysunek 2.27: Hierarchia klas ograniczeń



Rysunek 2.28: Współpraca obiektów klas optymalizacji

współpracy obiektów optymalizacji. Funkcja celu, zmienne projektowe oraz ograniczenia stanowią wraz z obiektem optymalizacji strukturę całość-część. Ponadto optymalizacja współpracuje z obiektem analizy MES uruchamiając analizę konstrukcji.

Nie stwarza większych problemów uogólnienie przedstawionych powyżej schematów optymalizacji i niezawodności tak, aby było możliwe rozwiązywanie problemów optymalizacji niezawodnościowej. W tym celu należy zadeklarować nową klasę funkcji celu, którą będzie prawdopodobieństwo awarii oraz nową klasę ograniczenia niezawodnościowego. Zmieni się też schemat współpracy obiektów ponieważ, obiekt optymalizacji będzie musiał uruchamiać obliczenia niezawodnościowe. Optymalizacja niezawodnościowa jest jednym z elementów planu rozwoju systemu w przyszłości.

Zaprezentowane w tym rozdziale zastosowanie podejścia obiektowego do analizy MES, optymalizacji i niezawodności pokazuje, że jest możliwe w relatywnie prosty sposób zbudowanie złożonych struktur danych opisujących bogactwo typów analiz, elementów skończonych czy materiałów. Implementacja strukturalna na tym poziomie ogólności byłaby zapewne znacznie bardziej skomplikowana i wymagała pracy kilku osób oraz poświęcenia znacznie więcej czasu. Klarowne struktury obiektowe pozwalają jednej osobie na tworzenie i utrzymanie tego systemu, a także stwarzają interesującą perspektywę rozwoju systemu w przyszłości. Ze względu na dużą różnorodność analiz, typów elementów, wyczerpująca prezentacja możliwości programu wykracza poza ramy tej pracy. W dalszej części pracy kilka przykładów numerycznych zilustruje zatem tylko wybrane zagadnienia.

Zagadnienia termolepkosprężystości

3.1 Wstęp

W tym rozdziale przedstawione będą zagadnienia składające się na model termolepkosprężysty. Po kilku wstępnych informacjach zaprezentowane zostaną podstawowe modele materiałów reologicznych. Następnie bardziej szczegółowo omówiony zostanie wybrany termolepkosprężysty model materiału. W termolepkosprężystym modelu własności mechaniczne materiału w danym punkcie zależne są od temperatury w tym punkcie. Istotnym elementem będzie również wyprowadzenie równań metody elementów skończonych dla termolepkosprężystego modelu materiału.

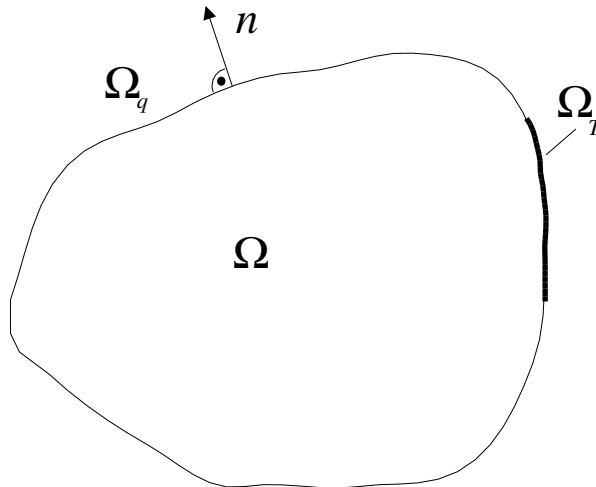
3.2 O reologii

Reologia jest nauką zajmującą się rozpatrywaniem własności mechanicznych materiałów konstrukcyjnych, w których wraz z upływem czasu obserwuje się zmiany stanu obejmujące również zjawisko dysypacji energii. Protoplastą reologii był przedstawiciel Jońskiej szkoły Heraklit z Efezu, który w swym dziele “O przyrodzie” zawarł słynne powiedzenie Παντα ρει - *panta rej* - wszystko płynie, powiedzenie będące istotą reologii.

Materiały lepkosprężyste to takie materiały, które wykazują jednocześnie cechy cieczy i ciał stałych. Ciała lepkosprężyste wykazują zdolność do akumulowania części energii odkształcenia podczas gdy reszta energii ulega rozproszeniu (*dysypacji*). Dysypacja energii spowodowana jest zjawiskiem zwanym *tarcie wewnętrzne*. Przez tarcie wewnętrzne rozumiemy opór, jaki stawiają przesuwane względem siebie warstwy cieczy. Dysypowana energia odkształcenia równa jest zatem pracy tarcia wewnętrznego. W ciele lepkosprężystym obciążonym stałym w czasie obciążeniem obserwujemy zmienny w czasie stan odkształcenia. Mówimy wówczas, że ciało płynie. Proces ten nosi też nazwę procesu *pełzania*. Innym charakterystycznym dla ciał lepkosprężystych zjawiskiem jest *relaksacja*. Pod pojęciem relaksacji rozumiemy zmieniający się w czasie stan naprężenia w próbce poddanej stałemu w czasie stanowi odkształcenia.

3.3 Niestacjonarny przepływ ciepła

3.3.1 Równanie przepływu ciepła



Rysunek 3.1: Ciało materialne z warunkami brzegowymi

Równanie różniczkowe (3.1) opisuje problem niestacjonarnego przepływu ciepła w ośrodkach termicznie ortotropowych,

$$\frac{\partial}{\partial x_1} \left[\lambda_1 \frac{\partial \theta}{\partial x_1} \right] + \frac{\partial}{\partial x_2} \left[\lambda_2 \frac{\partial \theta}{\partial x_2} \right] + \frac{\partial}{\partial x_3} \left[\lambda_3 \frac{\partial \theta}{\partial x_3} \right] + \dot{q} = \rho c \frac{\partial \theta}{\partial \tau}, \quad (3.1)$$

gdzie $\theta = T - T_0$, zaś T_0 jest temperaturą odniesienia na obszarze Ω (Rys. 3.3.1). W ogólnym przypadku równanie to jest równaniem nieliniowym. W pracy ograniczymy się do rozpatrywania liniowego przepływu ciepła, tzn. własności termiczne ośrodka opisywane będą niezależnymi od temperatury stałymi materiałowymi ($\lambda_1, \lambda_2, \lambda_3$ - wartości przewodności cieplnej, c - ciepło właściwe, ρ - gęstość materiału). Równanie (3.1) uzupełniają następujące warunki brzegowe:

$$\begin{aligned} \theta &= \hat{\theta}, & \mathbf{x} \in \partial\Omega_T, \\ \lambda_1 n_1 \frac{\partial \theta}{\partial x_1} + \lambda_2 n_2 \frac{\partial \theta}{\partial x_2} + \lambda_3 n_3 \frac{\partial \theta}{\partial x_3} &= \hat{q}, & \mathbf{x} \in \partial\Omega_q, \end{aligned} \quad (3.2)$$

3.3.2 Sformułowanie MES problemu niestacjonarnego przepływu ciepła

Do wyprowadzenia równań metody elementów skończonych wykorzystana zostanie metoda Galerkinia. Wprowadzamy aproksymację pola temperatury

$$\theta(\mathbf{x}, \tau) = \mathbf{N}(\mathbf{x})\boldsymbol{\theta}(\tau), \quad (3.3)$$

gdzie \mathbf{N} jest macierzą funkcji kształtu, a $\boldsymbol{\theta}$ jest wektorem temperatur węzłowych o wymiarze \tilde{N} , równym liczbie węzłów. Po podstawieniu tej aproksymacji do równań (3.1) oraz

(3.2) mamy

$$\begin{aligned} \left[\frac{\partial}{\partial x_1} \left[\lambda_1 \frac{\partial \mathbf{N}}{\partial x_1} \right] + \frac{\partial}{\partial x_2} \left[\lambda_2 \frac{\partial \mathbf{N}}{\partial x_2} \right] + \frac{\partial}{\partial x_3} \left[\lambda_3 \frac{\partial \mathbf{N}}{\partial x_3} \right] \right] \boldsymbol{\theta} + \dot{g} - \rho c \frac{\partial \theta}{\partial \tau} &= R_1, \quad \mathbf{x} \in \partial\Omega \\ - \left[\lambda_1 n_1 \frac{\partial \mathbf{N}}{\partial x_1} + \lambda_2 n_2 \frac{\partial \mathbf{N}}{\partial x_2} + \lambda_3 n_3 \frac{\partial \mathbf{N}}{\partial x_3} \right] \boldsymbol{\theta} + \hat{q} &= R_2, \quad \mathbf{x} \in \partial\Omega_q \end{aligned} \quad (3.4)$$

Ze względu na przybliżony charakter aproksymacji (3.3) residua R_1, R_2 są różne od zera. Minimalizując je otrzymujemy

$$\int_{\Omega} \mathbf{N}^T R_1 \, d\Omega + \int_{\Omega} \mathbf{N}^T R_2 \, d(\partial\Omega) = 0, \quad (3.5)$$

a po podstawieniu mamy

$$\begin{aligned} \left(\int_{\Omega} \left[\lambda_1 \frac{\partial \mathbf{N}^T}{\partial x_1} \frac{\partial \mathbf{N}}{\partial x_1} + \lambda_2 \frac{\partial \mathbf{N}^T}{\partial x_2} \frac{\partial \mathbf{N}}{\partial x_2} + \lambda_3 \frac{\partial \mathbf{N}^T}{\partial x_3} \frac{\partial \mathbf{N}}{\partial x_3} \right] d\Omega \right) \boldsymbol{\theta} - \int_{\Omega} \mathbf{N}^T \dot{g} \, d\Omega \\ + \left[\int_{\Omega} \rho c \mathbf{N}^T \mathbf{N} \, d\Omega \right] \dot{\boldsymbol{\theta}} - \int_{\Omega} \mathbf{N}^T \hat{q} \, d(\partial\Omega) = 0. \end{aligned} \quad (3.6)$$

Wprowadzając następujące oznaczenia

$$\mathbf{C} = \int_{\Omega} \rho c \mathbf{N}^T \mathbf{N} \, d\Omega, \quad (3.7)$$

$$\mathbf{K} = \int_{\Omega} \left[\lambda_1 \frac{\partial \mathbf{N}^T}{\partial x_1} \frac{\partial \mathbf{N}}{\partial x_1} + \lambda_2 \frac{\partial \mathbf{N}^T}{\partial x_2} \frac{\partial \mathbf{N}}{\partial x_2} + \lambda_3 \frac{\partial \mathbf{N}^T}{\partial x_3} \frac{\partial \mathbf{N}}{\partial x_3} \right] d\Omega, \quad (3.8)$$

$$\mathbf{P} = \int_{\Omega} \mathbf{N}^T \dot{g} \, d\Omega + \int_{\Omega} \mathbf{N}^T \hat{q} \, d(\partial\Omega), \quad (3.9)$$

zwane odpowiednio macierzą pojemności cieplnej \mathbf{C} , macierzą przewodnictwa \mathbf{K} oraz wektorem “obciążenia” (tu strumienia ciepła) \mathbf{P} , równanie (3.6) ma teraz następującą postać

$$\mathbf{C} \dot{\boldsymbol{\theta}} + \mathbf{K} \boldsymbol{\theta} = \mathbf{P}. \quad (3.10)$$

$\tilde{N} \times \tilde{N}$ wymiarowe macierze \mathbf{K} and \mathbf{C} są w ogólnym przypadku zależne od temperatury $\boldsymbol{\theta}$. Wektor “obciążeń” \mathbf{P} jest \tilde{N} wymiarowy.

By rozwiązać układ równań (3.10) zastosujemy schemat przyrostowy, zastępując wielkości $\boldsymbol{\theta}$ i $\dot{\boldsymbol{\theta}}$ wyrażeniami

$$\begin{aligned} {}^{t+\alpha\Delta t} \boldsymbol{\theta} &= (1 - \alpha) {}^t \boldsymbol{\theta} + \alpha {}^{t+\Delta t} \boldsymbol{\theta}, \quad \alpha \in [0, 1] \\ {}^{t+\alpha\Delta t} \dot{\boldsymbol{\theta}} &= \frac{1}{\Delta t} ({}^{t+\Delta t} \boldsymbol{\theta} - {}^t \boldsymbol{\theta}). \end{aligned} \quad (3.11)$$

Podstawiając te wyrażenia do równania (3.10) w chwili $t + \Delta t$, otrzymujemy

$$\begin{aligned} \left[\frac{1}{\Delta t} {}^{t+\alpha\Delta t}\mathbf{C} + \alpha {}^{t+\alpha\Delta t}\mathbf{K} \right] {}^{t+\Delta t}\boldsymbol{\theta} &= {}^{t+\Delta t}\mathbf{P} + \\ &+ \left[\frac{1}{\Delta t} {}^{t+\alpha\Delta t}\mathbf{C} - (1 - \alpha) {}^{t+\alpha\Delta t}\mathbf{K} \right] {}^t\boldsymbol{\theta}, \end{aligned} \quad (3.12)$$

lub

$${}^{t+\alpha\Delta t}\widehat{\mathbf{K}} {}^{t+\Delta t}\boldsymbol{\theta} = {}^{t+\alpha\Delta t}\widehat{\mathbf{P}}, \quad (3.13)$$

gdzie

$${}^{t+\alpha\Delta t}\widehat{\mathbf{K}} = \left[\frac{1}{\Delta t} {}^{t+\alpha\Delta t}\mathbf{C} + \alpha {}^{t+\alpha\Delta t}\mathbf{K} \right], \quad (3.14)$$

$${}^{t+\alpha\Delta t}\widehat{\mathbf{P}} = {}^{t+\alpha\Delta t}\mathbf{P} + \left[\frac{1}{\Delta t} {}^{t+\alpha\Delta t}\mathbf{C} - (1 - \alpha) {}^{t+\alpha\Delta t}\mathbf{K} \right] {}^t\boldsymbol{\theta}. \quad (3.15)$$

Równanie (3.13) może być rozwiązane ze względu na ${}^{t+\Delta t}\boldsymbol{\theta}$, przy zastosowaniu dowolnej metody iteracyjnej, przy czym ${}^t\boldsymbol{\theta}$ jest znane, podczas gdy macierze ${}^{t+\alpha\Delta t}\mathbf{C}$ i ${}^{t+\alpha\Delta t}\mathbf{K}$ obliczane są przy pomocy ostatniej dostępnej wartości wektora temperatury ${}^{t+\Delta t}\boldsymbol{\theta}$. W pracy ograniczymy się do liniowej wersji równania (3.13), w której ${}^{t+\alpha\Delta t}\mathbf{C}$ i ${}^{t+\alpha\Delta t}\mathbf{K}$ są macierzami niezależnymi od temperatury. W tym przypadku iteracja na kroku czasowym jest niepotrzebna.

3.4 Termolepkosprężystość

3.4.1 Model konstytutywny

Ogólna postać równania konstytutywnego w trójosiowym stanie naprężenia ma postać (3.16)

$$\sigma_{ij}(\tau) = \int_0^\tau C_{ijkl}(\xi - \xi') \frac{d}{d\tau'} [\varepsilon_{kl}(\tau') - \varepsilon_{kl}^{(\theta)}(\tau')] d\tau', \quad (3.16)$$

gdzie τ jest czasem, a ξ zredukowanym czasem, zdefiniowanym w następujący sposób

$$\xi = \xi(\tau) = \int_0^\tau \frac{d\zeta}{A(\theta(\zeta))}. \quad (3.17)$$

ξ' wartością funkcji ξ w chwili τ'

$$\xi' = \xi(\tau') = \int_0^{\tau'} \frac{d\zeta}{A(\theta(\zeta))}. \quad (3.18)$$

$A(\theta)$ jest współczynnikiem czasu relaksacji wprowadzającym zależność własności mechanicznych ciała od temperatury, $\varepsilon_{ij}^{(\theta)}$; to odkształcenia wstępne termiczne związane z rozszerzalnością cieplną materiału

$$\varepsilon_{ij}^{(\theta)}(\tau) = \int_0^\tau \alpha[\theta(\zeta)] \dot{\theta}[\theta(\zeta)] \delta_{ij} d\zeta, \quad (3.19)$$

gdzie $\alpha(\theta)$ jest chwilowym współczynnikiem rozszerzalności cieplnej zależnym w ogólności od temperatury, θ to absolutna temperatura ciała, $\theta_0 = \theta(0)$ temperatura odniesienia. Dla $\alpha(\theta) = \text{const}$ otrzymujemy

$$\varepsilon_{ij}^{(\theta)}(\tau) = \alpha[\theta(\tau) - \theta_0]\delta_{ij}. \quad (3.20)$$

Funkcja $C_{ijkl}(\xi)$ jest malejącą funkcją modułów relaksacji(3.16). Moduł ten oparty jest na zasadzie superpozycji, według której całkowite naprężenie w czasie ξ jest sumą przyrostów naprężeń w krokach czasowych $\xi' < \xi$. Statyka liniowa jest przypadkiem granicznym, gdy C_{ijkl} nie zależą od czasu.

Izotropowa wersja równania konstytutywnego ma postać

$$\begin{aligned} C_{ijkl}(\xi) &= \lambda(\xi)\delta_{ij}\delta_{kl} + \mu(\xi)(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) = \\ &= \left[\kappa(\xi) - \frac{2}{3}\mu(\xi) \right] \delta_{ij}\delta_{kl} + \mu(\xi)(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \end{aligned} \quad (3.21)$$

Zależności dla części kulistej i dewiatorowej mają postać

$$\begin{aligned} \sigma_{ij}^D(\tau) &= \int_0^\tau 2\mu(\xi(\tau) - \xi'(\tau')) \frac{d}{d\tau'} \varepsilon_{ij}^D(\tau') d\tau', \\ \sigma_{kk}(\tau) &= \int_0^\tau 3\kappa(\xi(\tau) - \xi'(\tau')) \frac{d}{d\tau'} \left[\varepsilon_{kk}(\tau') - \varepsilon_{kk}^{(\theta)}(\tau') \right] d\tau'. \end{aligned} \quad (3.22)$$

Moduły relaksacji dla ścinania i ściskania są zdefiniowane następująco

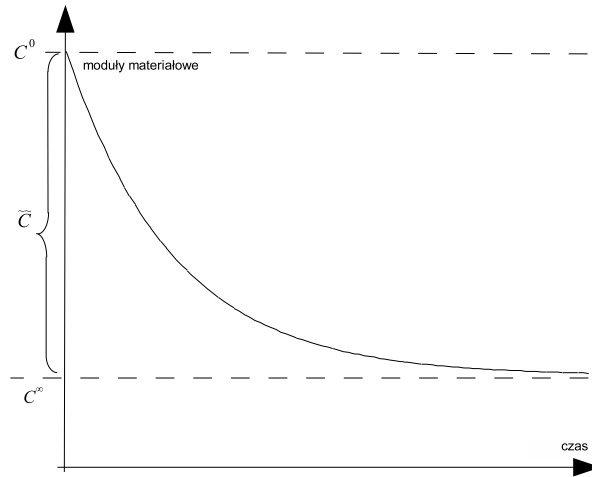
$$\begin{aligned} \mu(\tau) &= \mu^\infty + \tilde{\mu} \exp \left[-\frac{\xi(\tau)}{\varrho_\mu} \right], \\ \kappa(\tau) &= \kappa^\infty + \tilde{\kappa} \exp \left[-\frac{\xi(\tau)}{\varrho_\kappa} \right], \end{aligned} \quad (3.23)$$

gdzie dane parametry ϱ_μ i ϱ_κ są czasami relaksacji. Stąd

$$\begin{aligned} C_{ijkl}(\tau) &= C_{ijkl}^\infty + \tilde{C}_{ijkl}^\mu \exp \left[-\frac{\xi(\tau)}{\varrho_\mu} \right] + \tilde{C}_{ijkl}^\kappa \exp \left[-\frac{\xi(\tau)}{\varrho_\kappa} \right], \\ C_{ijkl}^\infty &= \left[\kappa^\infty - \frac{2}{3}\mu^\infty \right] \delta_{ij}\delta_{kl} + \mu^\infty(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \\ \tilde{C}_{ijkl}^\mu &= -\frac{2}{3}\tilde{\mu}\delta_{ij}\delta_{kl} + \tilde{\mu}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \\ \tilde{C}_{ijkl}^\kappa &= -\tilde{\kappa}\delta_{ij}\delta_{kl}. \end{aligned} \quad (3.24)$$

Definiujemy dla $\tau = 0$ moduły (ang. glassy moduli)

$$\begin{aligned} \mu(0) &\stackrel{\text{Df}}{=} \mu^0 = \mu^\infty + \tilde{\mu}, \\ \kappa(0) &\stackrel{\text{Df}}{=} \kappa^0 = \kappa^\infty + \tilde{\kappa}, \end{aligned} \quad (3.25)$$



Rysunek 3.2: Znaczenie modułów w modelu lepkosprężystym

oraz

$$C_{ijkl}(0) = C_{ijkl}^0 = C_{ijkl}^\infty + \tilde{C}_{ijkl}^\mu + \tilde{C}_{ijkl}^\kappa. \quad (3.26)$$

Dla przejrzystości ograniczymy rozważania do jednego czasu relaksacji, stąd (3.24)₁ będzie teraz miało postać

$$C_{ijkl}(\tau) = C_{ijkl}^\infty + \tilde{C}_{ijkl} \exp \left[-\frac{\xi(\tau)}{\varrho\kappa} \right]. \quad (3.27)$$

Dla $\tau = 0$, $\xi = 0$ moduły C_{ijkl} stają się modułami C_{ijkl}^0 (3.26), por. Rys. 3.2

$$C_{ijkl}(0) = C_{ijkl}^\infty + \tilde{C}_{ijkl}. \quad (3.28)$$

Stosując (3.27), (3.28) przekształcamy równanie konstytutywne (3.16) do postaci

$$\begin{aligned} \sigma_{ij}(\tau) &= \int_0^\tau \left[C_{ijkl}^\infty + \tilde{C}_{ijkl} \exp \left[-\frac{\xi - \xi'}{\varrho} \right] \right] \frac{d}{dt'} \left[\varepsilon_{ij}(\tau') - \varepsilon_{ij}^{(\theta)}(\tau') \right] d\tau' = \\ &= \int_0^\tau \left[C_{ijkl}^0 - \tilde{C}_{ijkl} \left[1 - \exp \left[-\frac{\xi - \xi'}{\varrho} \right] \right] \right] \frac{d}{dt'} \left[\varepsilon_{ij}(\tau') - \varepsilon_{ij}^{(\theta)}(\tau') \right] d\tau' = \\ &= C_{ijkl}^0 [\varepsilon_{kl}(\tau) - \varepsilon_{kl}^{(\theta)}(\tau)] - \sigma_{ij}^{(c)}(\tau), \end{aligned} \quad (3.29)$$

gdzie fikcyjne “naprężenie pełzania” (różnica pomiędzy naprężeniem sprężystym $C_{ijkl}^0 [\varepsilon_{kl}(\tau) - \varepsilon_{kl}^{(\theta)}(\tau)]$ a aktualnym naprężeniem σ_{ij}) wynosi

$$\sigma_{ij}^{(c)}(\tau) = \int_0^\tau \tilde{C}_{ijkl} \left[1 - \exp \left[-\frac{\xi - \xi'}{\varrho} \right] \right] \frac{d}{dt'} \left[\varepsilon_{kl}(\tau') - \varepsilon_{kl}^{(\theta)}(\tau') \right] d\tau'. \quad (3.30)$$

Pole temperatury $\theta(x, \tau)$ otrzymane jest z rozwiązania problemu niestacjonarnego przepływu ciepła. Zakładamy, że na wartość temperatury nie mają wpływu własności mechaniczne materiału, natomiast zależą one od temperatury.

Dla wyznaczenia naprężeń σ_{ij} w aktualnej chwili czasowej τ na podstawie związku (3.29) należy:

- (a) rozwiązać problem przepływu ciepła (3.1), (3.2) w chwili $\theta(\tau)$
- (b) znaleźć odkształcenia termiczne $\varepsilon_{ij}^{(\theta)}$ w chwili τ z (3.19)
- (c) wyznaczyć zredukowany czas ξ w τ w/g (3.17)
- (d) podstawić wyznaczone wielkości do równania (3.16)

Box 1. Idea rozwiązania problemu termolepkosprężystego

Możliwe jest stworzenie numerycznego modelu w oparciu o równania konstytutywne (3.29), (3.30) - taka procedura mogłaby wymagać obliczania całek Volterry (3.30). Tutaj zastosowany zostanie schemat przyrostowy. Takie podejście jest szeroko stosowane w obliczeniach numerycznych. Dla znalezienia schematu przyrostowego zróżniczkujemy równanie (3.30) względem czasu

$$\begin{aligned} \frac{d\sigma_{ij}^{(c)}(\tau)}{d\tau} &= \int_0^\tau \left[\tilde{C}_{ijkl} \frac{1}{\rho A(\tau)} \exp \left[-\frac{\xi(\tau) - \xi'(\tau')}{\varrho} \right] \right] \frac{d\tilde{\varepsilon}_{kl}(\tau')}{d\tau'} d\tau' = \\ &= \frac{\tilde{C}_{ijkl}}{\rho A(\tau)} \int_0^\tau \left[1 - \exp \left[-\frac{\xi(\tau) - \xi'(\tau')}{\varrho} \right] \right] \frac{d\tilde{\varepsilon}_{kl}(\tau')}{d\tau'} d\tau' + \\ &+ \frac{\tilde{C}_{ijkl}}{\rho A(\tau)} \int_0^\tau \frac{d\tilde{\varepsilon}_{kl}(\tau')}{d\tau'} d\tau' = \frac{1}{\rho A(\tau)} \left[\tilde{C}_{ijkl} \tilde{\varepsilon}_{kl}(\tau) - \sigma_{ij}^{(c)}(\tau) \right], \end{aligned} \quad (3.31)$$

gdzie

$$\tilde{\varepsilon}_{ij} \stackrel{Df}{=} \varepsilon_{ij} - \varepsilon_{ij}^{(\theta)}. \quad (3.32)$$

Równanie (3.31) jest wygodną postacią ewolucji w czasie naprężeń pełzania $\sigma_{ij}^{(c)}$.

3.4.2 Sformułowanie MES

Równanie konstytutywne w postaci (3.29), (3.31) pozwala łatwo zastosować standardowe algorytmy przyrostowe metody elementów skończonych. Naprężenia $\sigma_{ij}^{(c)}$ potraktowane są jako wewnętrzne parametry, których ewolucję w czasie opisuje równanie (3.31).

Do rozwiązania równań MES problemu zastosowany zostanie schemat bezpośredniego całkowania równania. Zakładając, że zasady prac wirtualnych i temperatury wirtualnej są spełnione (z założoną dokładnością) na początku kroku czasowego $t \rightarrow t + \Delta t$ (gdzie mechaniczne i cieplne równania równowagi wraz z ich warunkami brzegowymi są spełnione w sensie całkowym w czasie t), skorzystanie z powyższych zasad dla $t = t + \Delta t$ daje możliwość iteracyjnego wyznaczania węzłowego przyrostu przemieszczeń i temperatury, potrzebnego do wyznaczenia naprężeń i innych wielkości dla $t + \Delta t$.

Dla rozwiązania mechanicznych równań problemu termolepkosprężystego potrzebne będą: schemat całkowania równania oraz konsystentna macierz sztywności. Bezpośrednie

całkowanie równania konstytutywnego będzie wykonywane w każdym punkcie, z zastosowaniem schematu Eulera do obliczenia wartości na końcu kroku $\sigma_{ij}^{(c)}$, z równań (3.31)

$$\frac{{}^{t+\Delta t}\boldsymbol{\sigma}^{(c)} - {}^t\boldsymbol{\sigma}^{(c)}}{\Delta t} = \frac{1}{\rho} \frac{1}{{}^{t+\Delta t}A} \left[\tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)} \right]. \quad (3.33)$$

Po wyznaczeniu ${}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}$ mamy

$${}^{t+\Delta t}\boldsymbol{\sigma}^{(c)} = \left[\frac{1}{\Delta t} + \frac{1}{{}^{t+\Delta t}A\rho} \right]^{-1} \left[\frac{{}^t\boldsymbol{\sigma}^{(c)}}{\Delta t} + \frac{\tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{{}^{t+\Delta t}A\rho} \right]. \quad (3.34)$$

Równanie konstytutywne (3.16) można zapisać w postaci

$${}^{t+\Delta t}\boldsymbol{\sigma} = \mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}. \quad (3.35)$$

Analizując równania (3.34), (3.35) widzimy, że przyrost odkształcenia ${}^{t+\Delta t}\boldsymbol{\varepsilon}$ jest obliczany raz i iteracja nie jest potrzebna na poziomie punktu dla obliczenia wielkości ${}^{t+\Delta t}\boldsymbol{\sigma}$. Równanie konstytutywne (3.35) można napisać w postaci

$$\begin{aligned} {}^{t+\Delta t}\boldsymbol{\sigma} &= \mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - \left[\frac{1}{\Delta t} + \frac{1}{{}^{t+\Delta t}A\rho} \right]^{-1} \left[\frac{{}^t\boldsymbol{\sigma}^{(c)}}{\Delta t} + \frac{\tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{{}^{t+\Delta t}A\rho} \right] = \\ &= {}^{t+\Delta t}\mathbf{C}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - \frac{{}^{t+\Delta t}A\rho}{\Delta t + {}^{t+\Delta t}A\rho} {}^t\boldsymbol{\sigma}^{(c)}, \end{aligned} \quad (3.36)$$

gdzie

$$\begin{aligned} {}^{t+\Delta t}\mathbf{C}^* &= \frac{\partial {}^{t+\Delta t}\boldsymbol{\sigma}}{\partial {}^{t+\Delta t}\boldsymbol{\varepsilon}} = \mathbf{C}^0 - \left[\frac{1}{\Delta t} + \frac{1}{{}^{t+\Delta t}A\rho} \right]^{-1} \frac{\tilde{\mathbf{C}}}{{}^{t+\Delta t}A\rho} = \\ &= \mathbf{C}^0 - \frac{\Delta t \tilde{\mathbf{C}}}{\Delta t + {}^{t+\Delta t}A\rho} \end{aligned} \quad (3.37)$$

jest stycznym operatorem konsystentnym ze schematem całkowania równania.

Globalne równanie równowagi zapisane w chwili $t + \Delta t$ w postaci zasady prac wirtualnych ma postać

$$\int_{\Omega} {}^{t+\Delta t}\boldsymbol{\sigma}^T \delta \boldsymbol{\varepsilon} d\Omega = \int_{\Omega} {}^{t+\Delta t}\hat{\mathbf{f}}^T \delta \mathbf{u} d\Omega + \int_{\partial\Omega_{\sigma}} \hat{\mathbf{t}}^T \delta \mathbf{u} d\partial\Omega_{\sigma}, \quad (3.38)$$

gdzie $\hat{\mathbf{f}}$ i $\hat{\mathbf{t}}$ są objętościowymi i brzegowymi wektorami obciążenia działającego na ciało. Po wprowadzeniu w równaniu (3.38) standardowej aproksymacji pola przemieszczeń stosowanej w metodzie elementów mamy

$$\begin{aligned} \mathbf{u} &= \mathbf{N}\mathbf{q}, & \Delta \mathbf{u} &= \mathbf{N}\Delta \mathbf{q}, \\ \boldsymbol{\varepsilon} &= \mathbf{L}\mathbf{u} = \mathbf{L}\mathbf{N}\mathbf{q} = \mathbf{B}\mathbf{q}, & \boldsymbol{\theta} &= \tilde{\mathbf{N}}\boldsymbol{\theta}, \\ \mathbf{B} &= \mathbf{L}\mathbf{N} \end{aligned} \quad (3.39)$$

i podstawieniu do (3.36) otrzymujemy

$$\begin{aligned}
& ({}^i\mathbf{q} + \Delta\mathbf{q})^T \left[\int_{\Omega} \mathbf{B}^T \mathbf{C}^0 \mathbf{B} d\Omega \right] \delta\mathbf{q} - \left[\int_{\Omega} \frac{{}^{t+\Delta t}A\varrho}{{\Delta t + \varrho} \, {}^{t+\Delta t}A} {}^i\boldsymbol{\sigma}^{(c)T} \mathbf{B} d\Omega \right] \delta\mathbf{q} + \\
& - ({}^t\mathbf{q} + \Delta\mathbf{q})^T \left[\int_{\Omega} \mathbf{B}^T \frac{\Delta t}{{\Delta t + \varrho} \, {}^{t+\Delta t}A} \tilde{\mathbf{C}} \mathbf{B} d\Omega \right] \delta\mathbf{q} - \left[\int_{\Omega} \boldsymbol{\alpha}^T \theta_0 {}^{t+\Delta t} \mathbf{C}^{*T} \mathbf{B} d\Omega \right] \delta\mathbf{q} + \quad (3.40) \\
& \boldsymbol{\theta}^T \left[\int_{\Omega} \bar{\mathbf{N}}^T \boldsymbol{\alpha}^T {}^{t+\Delta t} \mathbf{C}^{*T} \mathbf{B} d\Omega \right] \delta\mathbf{q} = \left[\int_{\Omega} \rho ({}^t\mathbf{f} + \Delta\mathbf{f})^T \mathbf{N} d\Omega \right] \delta\mathbf{q} + \\
& \left[\int_{\partial\Omega_{\sigma}} ({}^t\mathbf{t} + \Delta\mathbf{t})^T \mathbf{N} d\partial\Omega_{\sigma} \right] \delta\mathbf{q}.
\end{aligned}$$

Dla zwięzłości zapisu wprowadzamy następujące oznaczenia

$${}^{t+\Delta t} \tilde{D} = \frac{\Delta t}{{\Delta t + \varrho} \, {}^{t+\Delta t}A\varrho}, \quad {}^{t+\Delta t} D^{(c)} = \frac{{}^{t+\Delta t}A\varrho}{{\Delta t + \varrho} \, {}^{t+\Delta t}A\varrho}. \quad (3.41)$$

Wszystkie wielkości (zamieszczone poniżej), zależne od temperatury zawsze zależą od ${}^{t+\Delta t}\boldsymbol{\theta}$ i nigdy nie zależą od ${}^t\boldsymbol{\theta}$. Dlatego nie będzie używany indeks w wielkościach zależnych od temperatury w $t + \Delta t$. Każda wielkość z indeksami t , $t + \Delta t$ lub bez indeksu będzie traktowana jak zależna od ${}^{t+\Delta t}\boldsymbol{\theta}$. Symbole (3.41) będą stosowane jako \tilde{D} , $D^{(c)}$. Natomiast indeksy w równaniach przepływu ciepła są stosowane bez zmian.

$$\mathbf{K}^0 = \int_{\Omega} \mathbf{B}^T \mathbf{C}^0 \mathbf{B} d\Omega, \quad (3.42)$$

$$\tilde{\mathbf{K}} = \left[\int_{\Omega} \tilde{D} \mathbf{B}^T \tilde{\mathbf{C}} \mathbf{B} d\Omega \right], \quad (3.43)$$

$$\mathbf{K}^* = \int_{\Omega} \mathbf{B}^T \mathbf{C}^* \mathbf{B} d\Omega = \mathbf{K}^0 - \tilde{\mathbf{K}}, \quad (3.44)$$

$$\mathbf{K}_S = \int_{\Omega} \bar{\mathbf{N}}^T \boldsymbol{\alpha}^T \mathbf{C}^* \mathbf{B} d\Omega, \quad (3.45)$$

$${}^{t+\Delta t} \mathbf{Q} = \left[\int_{\Omega} ({}^t\hat{\mathbf{f}} + \Delta\hat{\mathbf{f}})^T \mathbf{N} d\Omega \right] + \left[\int_{\partial\Omega_{\sigma}} ({}^t\hat{\mathbf{t}} + \Delta\hat{\mathbf{t}})^T \mathbf{N} d\partial\Omega_{\sigma} \right], \quad (3.46)$$

$${}^t \mathbf{F}^0 = \left[\int_{\Omega} \mathbf{B}^T \mathbf{C}^0 \mathbf{B} d\Omega \right] {}^t \mathbf{q}, \quad (3.47)$$

$$\mathbf{F}^{(\theta)} = \int_{\Omega} \boldsymbol{\alpha}^T \theta_0 \mathbf{C}^* \mathbf{B} d\Omega, \quad (3.48)$$

$${}^t \tilde{\mathbf{F}} = \left[\int_{\Omega} \tilde{D} \mathbf{B}^T \tilde{\mathbf{C}} \mathbf{B} d\Omega \right] {}^t \mathbf{q} = \tilde{\mathbf{K}} {}^t \mathbf{q}, \quad (3.49)$$

$$\Delta \tilde{\mathbf{F}} = \left[\int_{\Omega} \tilde{D} \mathbf{B}^T \tilde{\mathbf{C}} \mathbf{B} d\Omega \right] \Delta \mathbf{q} = \tilde{\mathbf{K}} \Delta \mathbf{q}, \quad (3.50)$$

$${}^t \tilde{\mathbf{F}}^{(c)} = \left[\int_{\Omega} D^{(c)} \mathbf{B}^T {}^t \boldsymbol{\sigma}^{(c)} d\Omega \right], \quad (3.51)$$

$${}^t \mathbf{F}^{(c)} = {}^t \tilde{\mathbf{F}} + \tilde{\mathbf{F}}^{(c)}, \quad (3.52)$$

$${}^{t+\Delta t} \hat{\mathbf{Q}} = {}^{t+\Delta t} \mathbf{Q} - {}^t \mathbf{F}^0 + {}^t \mathbf{F}^{(c)} + \mathbf{F}^{(\theta)}, \quad (3.53)$$

ze względu na symetrię macierzy $\mathbf{K}^0, \tilde{\mathbf{K}}, \mathbf{K}^*$ oraz dowolność $\Delta \mathbf{q}$ można zapisać równanie (3.40) jako

$$\left[\mathbf{K}^0 - \tilde{\mathbf{K}} \right] \Delta \mathbf{q} = {}^{t+\Delta t} \mathbf{Q} - {}^t \mathbf{F}^0 + {}^t \mathbf{F}^{(c)} + \mathbf{F}^{(\theta)}, \quad (3.54)$$

lub bardziej zwięźle

$${}^{t+\Delta t} \mathbf{K}^* \Delta \mathbf{q} = {}^{t+\Delta t} \hat{\mathbf{Q}}. \quad (3.55)$$

Układ równań dla problemu termolepkosprężystego ma postać

$$\begin{bmatrix} {}^{t+\alpha\Delta t} \hat{\mathbf{K}} & 0 \\ \mathbf{K}_S & \mathbf{K}^* \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \boldsymbol{\theta} \\ \Delta \mathbf{q} \end{bmatrix} = \begin{bmatrix} {}^{t+\alpha\Delta t} \hat{\mathbf{P}} \\ {}^{t+\Delta t} \hat{\mathbf{Q}} \end{bmatrix}. \quad (3.56)$$

Układ ten składa się z dwóch podukładów równań, które mogą być rozwiązywane oddzielnie, jeden po drugim. Rozpatrując liniowy przepływ ciepła macierze ${}^{t+\alpha\Delta t} \hat{\mathbf{K}}$ stają się niezależne od czasu $\hat{\mathbf{K}}$; pierwszy podukład może być wyodrębniony z układu równań (4.17) w postaci

$$\hat{\mathbf{K}} {}^{t+\Delta t} \boldsymbol{\theta} = {}^{t+\alpha\Delta t} \hat{\mathbf{P}}. \quad (3.57)$$

Mając wyznaczoną temperaturę ${}^{t+\Delta t} \boldsymbol{\theta}$ można obliczyć macierz styczną oraz wektor prawych stron dla drugiego podukładu z (4.17)

$$\mathbf{K}^* \Delta \mathbf{q} = {}^{t+\Delta t} \hat{\mathbf{Q}} - \mathbf{K}_S {}^{t+\Delta t} \boldsymbol{\theta}. \quad (3.58)$$

Po rozwiązaniu otrzymamy przyrost przemieszczeń $\Delta \mathbf{q}$ w rozpatrywanym kroku czasowym. Drugi układ równań (3.58) może być zapisany w innej formie przez przeniesienie na prawą stronę wielkości $\tilde{\mathbf{K}} \Delta \mathbf{q}$ w równaniu (3.54). Otrzymamy zatem

$$\mathbf{K}^0 \Delta \mathbf{q} = {}^{t+\Delta t} \hat{\mathbf{Q}} - \mathbf{K}_S {}^{t+\Delta t} \boldsymbol{\theta} - \tilde{\mathbf{K}} \Delta \mathbf{q}. \quad (3.59)$$

Stosując oznaczenie (3.51) mamy

$$\mathbf{K}^0 \Delta \mathbf{q} = {}^{t+\Delta t} \widehat{\mathbf{Q}} - \mathbf{K}_S {}^{t+\Delta t} \boldsymbol{\theta} - \Delta \widetilde{\mathbf{F}}. \quad (3.60)$$

Zauważmy, że zastosowanie równania (3.55) redukuje procedure do jednokrotnego rozwiązania liniowego układu równań algebraicznych w każdym przyroście czasowym. Natomiast procedura oparta na równaniu (3.59) wymaga iteracji w/g wzoru

$$\mathbf{K}^0 \Delta \mathbf{q}^{(i+1)} = {}^{t+\Delta t} \widehat{\mathbf{Q}} - \mathbf{K}_S {}^{t+\Delta t} \boldsymbol{\theta} - \widetilde{\mathbf{K}} \Delta \mathbf{q}^{(i)}, \quad i = 1, 2, \dots \quad (3.61)$$

Warto zauważyć, że macierz \mathbf{K}^0 jest stała w całym procesie, podczas gdy macierz \mathbf{K}^* jest stała tylko w przypadku stałego w czasie rozkładu temperatury. W przeciwnym przypadku macierz musi być obliczana i dekomponowana na każdym kroku czasowym. Dlatego też nie można zakładać, że rozwiązanie oparte na równaniu (3.58) będzie zawsze efektywniejsze. Może się okazać, że obliczenie i dekompozycja macierzy \mathbf{K}^* sprawi, że stosowanie podejścia (3.61) daje krótsze czasy obliczeń.

Prosty przykład numeryczny wraz z analizą wrażliwości zawarty został w następnym rozdziale.

3.5 Model zniszczenia

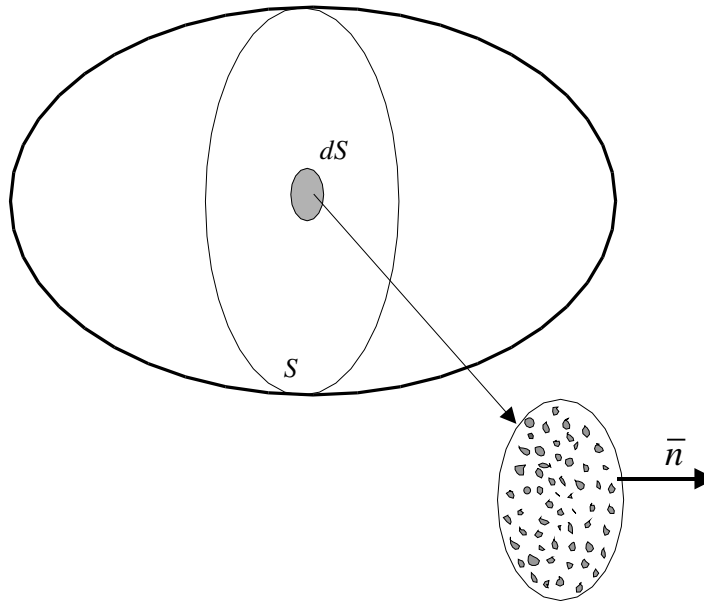
3.5.1 Wstęp

Odkąd człowiek zaczął budować konstrukcje inżynierskie, modelowanie ich zniszczenia stawało się coraz bardziej istotnym elementem procesu projektowania. Zniszczenia materiału od wieków interesowało badaczy. Wszechstronnie uzdolniony Leonardo da Vinci około roku 1500 zajmował się tym zjawiskiem i próbował je opisać poprzez zmienne mechaniczne. W ślad za nim poszło wielu innych znanych badaczy (m.in. Coulomb, Tresca, Huber-Mises, Mohr). Podawali oni wiele kryteriów zniszczenia będących funkcjami składowych tensora naprężenie czy odkształcenia. Jednak dopiero niedawno zaczęto modelować postępującą degradację ośrodka wynikającą z mikrostrukturalnych zmian poprzedzających całkowite, makroskopowe zniszczenie. Era współczesnego modelowania procesów zniszczenia rozpoczęła się w 1958 roku, kiedy to Katchanov opublikował pierwszą pracę, poświęconą ciąglemu parametrowi opisującemu postępujący proces zniszczenia materiału [57]. Od tego czasu pojawiło się również wiele innych prac poświęconych tej problematyce [63, 58, 60]. Koncepcja ta była następnie rozwijana, m.in. uogólniona na złożone stany naprężenia.

Zdefiniowanie zmiennych mechanicznych reprezentujących stan zniszczenia jest trudne. Makroskopowe obserwacje nie pozwalają odróżnić materiału zniszczonego od niezniszczonego. Dlatego bezpośredni, makroskopowy pomiar zniszczenia jest niemożliwy do wykonania. Istnieje kilka podejść pozwalających na tworzenie modeli zniszczenia. Jednym z nich są obserwacje mikrostrukturalne pozwalające na zaobserwowanie mikroszczelin. Obserwacje te prowadzą do stworzenia mikroskopowego modelu zniszczenia i na tej podstawie stworzenia modelu makroskopowego, czyli zidentyfikowania odpowiednich zmiennych opisujących to zjawisko. Inną możliwością jest obserwowanie i generalizacja danych pochodzących z eksperymentów. Zniszczenie materiału szacuje się po dokonaniu pomiarów różnych własności mechanicznych, elektrycznych, czy też akustycznych. Najłatwiejsze do

interpretacji, z punktu widzenia określenia zmiennych zniszczenia, są pomiary własności mechanicznych. W tej pracy zastosowane będzie to właśnie podejście, czyli wprowadzona przez Rabotnova koncepcja naprężeń efektywnych.

3.5.2 Koncepcja modelu zniszczenia



Rysunek 3.3: Ciało ulegające zniszczeniu

Rozważania na temat modelu zniszczenia przeprowadzimy na przykładzie bryły reprezentującej ciało ulegające zniszczeniu. Niech S będzie powierzchnią przekroju tej bryły, z wektorem normalnym do tej powierzchni \bar{n} . Wraz z postępującym zniszczeniem materiału, obserwujemy powiększanie się powierzchni mikropęknięć na przekroju materiału. Oznaczmy pole powierzchni przekroju próbki jako A , natomiast sumaryczne pole powierzchni mikropęknięć jako A_D . Polem powierzchni niezniszczonego materiału jest zatem różnica $A - A_D$. Zdefiniujemy teraz zmienną zniszczenia jako

$$d_n = \frac{A_D}{A}. \quad (3.62)$$

Zmienna zniszczenia jest zatem powierzchniowym rozkładem gęstości nieciągłości materiału w przekroju określonym przez wektor normalny \bar{n} . Wielkość ta przyjmuje charakterystyczne wartości:

- $d_n = 0$ materiał jest w pierwotnym, niezniszczonym stanie,
- $0 < d_n < 1$ materiał podlega procesowi zniszczenia, zmienna charakteryzuje stopień zniszczenia,
- $d_n = 1$ materiał całkowicie zniszczony.

Rozpatrujemy w tej pracy model zniszczenia izotropowego. Oznacza to, że w każdym punkcie ciała wielkość d jest jednakowa dla każdego przekroju. Stan zniszczenia w danym punkcie ciała opisuje zatem jedna skalarna wielkość $d_n = d$. W przypadku anizotropowego zniszczenia stan zniszczenia zależny byłby od wektora normalnego i do opisu stanu zniszczenia potrzebny byłby tensor drugiego rzędu.

Zdefiniujemy teraz naprężenia efektywne w stanie jednoosiowym jako

$$\sigma_a = \frac{P}{A - A_D} = \frac{P}{A(1 - d)} = \frac{\sigma}{(1 - d)}, \quad (3.63)$$

gdzie P jest siłą jaką obciążamy próbkę w stanie jednoosiowym a σ to naprężenia odniesione do powierzchni pierwotnej. Naprężenia efektywne są zatem odniesione do powierzchni niezniszczonego materiału. Zachodzi

- $\sigma_a = \sigma$ dla materiału w pierwotnym, niezniszczonym stanie,
- $\sigma_a \geq \sigma$ dla rozpoczętego procesu zniszczenia,
- $\sigma_a \rightarrow \infty$ w momencie całkowitego zniszczenia.

Jedną z najważniejszych zalet naprężeń efektywnych jest to, że wszelkie relacje konstytutywne są takie same, jak dla materiału w stanie pierwotnym, niezniszczonym z tym, że zwykle naprężenia zastępuje się naprężeniami efektywnymi. Dla prawa Hooke'a mamy na przykład

$$\frac{\sigma}{(1 - d)} = E \cdot \varepsilon, \quad (3.64)$$

czyli

$$\sigma = (1 - d)E \cdot \varepsilon, \quad (3.65)$$

gdzie E - moduł Younga.

Proces zniszczenia w danym punkcie materiału rozpoczyna się wtedy, gdy pewne wielkości mechaniczne przekroczą ustalony próg. Tym progiem są zwykle naprężenia lub odkształcenia graniczne. W przypadku jednoosiowego stanu naprężenia, jeśli naprężenia czy odkształcenia są poniżej wartości progowych, zniszczenie nie występuje.

$$\sigma_c < \sigma < \sigma_t \longrightarrow D = \dot{D} = 0, \quad (3.66)$$

gdzie σ_c , σ_t to wartości graniczne naprężenia w pręcie przy ściskaniu i przy rozciąganiu.

W przypadku złożonego stanu naprężenia proces zniszczenia rozpoczyna się z chwilą osiągnięcia powierzchni granicznej wyrażonej w naprężeniach

$$F(\sigma, d) = 0, \quad (3.67)$$

lub wyrażonej w odkształceniach

$$F(\varepsilon, d) = 0. \quad (3.68)$$

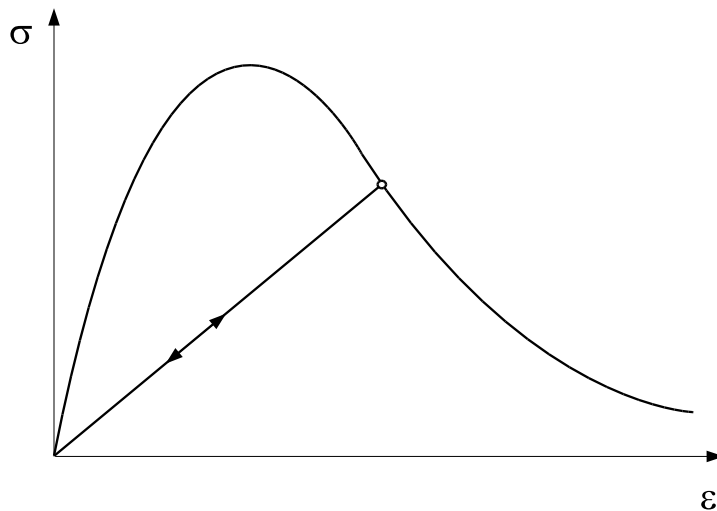
W rozpatrywanym modelu zniszczenia funkcja graniczna będzie miała postać

$$F(\varepsilon_{\text{eq}}, \kappa) = \varepsilon_{\text{eq}} - \kappa(\varepsilon_{\text{eq}}), \quad (3.69)$$

gdzie κ jest maksymalnym odkształceniem efektywnym w historii odkształcenia materiału

$$\kappa = \max(\varepsilon_{\text{eq}}, \kappa_i), \quad (3.70)$$

gdzie κ_i jest progiem inicjacji zniszczenia. Natomiast ε_{eq} jest funkcją niezmiennika stanu odkształcenia, zwanego odkształceniem efektywnym. Dla wartości funkcji granicznej $F < 0$ zniszczenie materiału nie następuje i materiał zachowuje się zgodnie ze swoimi pierwotnymi cechami (np. sprężystość). Gdy wartość funkcji granicznej zaczyna przyjmować wartości dodatnie, wówczas zachodzi proces zniszczenia. Dzięki takiej definicji wartości κ (3.70), wartości zmiennej zniszczenia nie zmniejszają się w procesie odciążenia materiału, co byłoby równoznaczne z “naprawianiem” się materiału nie mającym potwierdzenia w rzeczywistości. W tym modelu, zgodnie z rzeczywistym zachowaniem się materiałów w procesie odciążania, degradacja materiału pozostaje nieodwracalna i postępuje dalej z chwilą ponownego obciążenia materiału, co ilustruje rysunek 3.4. Postęp zniszczenia



Rysunek 3.4: Stagnacja zniszczenia w procesie odciążania próbki

odbywa się, zgodnie z warunkami Kuhna Tuckera

$$F < 0, \quad F\dot{\kappa} = 0, \quad \dot{\kappa} \geq 0. \quad (3.71)$$

Kształt powierzchni zniszczenia zależy od sposobu zdefiniowania odkształcenia efektywnego. Definicja odkształcenia efektywnego zależy od rodzaju modelowanego materiału. Poniżej podane zostanie kilka najbardziej znanych definicji.

- *Energia*

$$\varepsilon_{\text{eq}} = \frac{1}{2} \varepsilon^T \mathbf{D} \varepsilon, \quad (3.72)$$

takie odkształcenie efektywne jest używane w analizie termodynamiki procesów nieodwracalnych.

- *Znormalizowana energia*

$$\varepsilon_{\text{eq}} = \sqrt{\frac{1}{E} \varepsilon^T \mathbf{D} \varepsilon}. \quad (3.73)$$

- *Dodatnie odkształcenie główne*

$$\varepsilon_{\text{eq}} = \sqrt{\sum_{i=1}^3 \langle \varepsilon_i \rangle^2}, \quad (3.74)$$

gdzie $\langle \varepsilon_i \rangle = \frac{1}{2}(\varepsilon_i + |\varepsilon_i|)$ a ε_i to i -te odkształcenie główne.

- *Zmodyfikowana definicja Hubera-Misesa*

$$\varepsilon_{\text{eq}} = \frac{(\eta - 1)I_1}{2\eta(1 - 2\nu)} + \frac{1}{2\eta} \sqrt{\frac{(\eta - 1)^2 I_1^2}{(1 - 2\nu)^2} + \frac{2\eta J_2}{(1 + \nu)^2}}, \quad (3.75)$$

gdzie η to stosunek wytrzymałości na ściskanie do wytrzymałości na rozciąganie, a I_1 oraz J_2 to niezmienniki tensora odkształcenia zdefiniowane następująco

$$I_1 = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}, \quad (3.76)$$

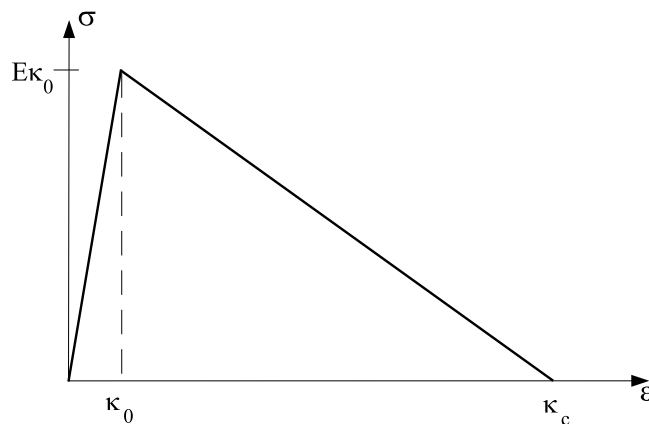
$$J_2 = (\varepsilon_{xx}^2 + \varepsilon_{yy}^2 + \varepsilon_{zz}^2 - \varepsilon_{xx}\varepsilon_{yy} - \varepsilon_{yy}\varepsilon_{zz} - \varepsilon_{zz}\varepsilon_{xx})/3 + \varepsilon_{xy}^2 + \varepsilon_{yz}^2 + \varepsilon_{xz}^2. \quad (3.77)$$

Wiele materiałów wykazuje liniowy przyrost odkształcenia, wraz ze wzrostem obciążenia, aż do pewnej granicy, zwanej granicą proporcjonalności. Po osiągnięciu granicy, powyżej której występuje zjawisko zniszczenia powolna degradacja materiału objawia się postępującą utratą zdolności materiału do przenoszenia obciążeń aż do jej całkowitego braku w chwili zniszczenia. Zjawisko to nazywa się *osłabieniem* materiału. Jest ono wyraźnie widoczne na wykresach siła-przemieszczenie obrazujących wyniki eksperymentów laboratoryjnych prób nośności Rys. 3.4. Słabnięcie materiału jest procesem postępującym wraz ze wzrostem zniszczenia zatem można mówić o ewolucji osłabienia. W opisywanym modelu zniszczenia, osłabienie postępuje wraz ze wzrostem odkształceń efektywnych, a więc prawo ewolucji osłabienia jest funkcją ε_{eq} . Istnieje kilka funkcji słabnięcia. Trzy najczęściej spotykane w literaturze zostaną opisane poniżej.

Oslabienie materiału jest zjawiskiem mikrostrukturalnym. Postępujące zniszczenie materiału spowodowane jest łączeniem się mikropeknięć w obszarze zniszczenia. Na pozostałym obszarze, gdzie zniszczenie nie występuje, obserwuje się nawet odciążenie materiału czyli zmniejszanie się odkształcenia. To zjawisko nazywa się *lokalizacją odkształceń*.

- *Oslabienie liniowe.* W liniowym osłabieniu ewolucja zmiennej zniszczenia zdefiniowana jest następująco (Rys. 3.5)

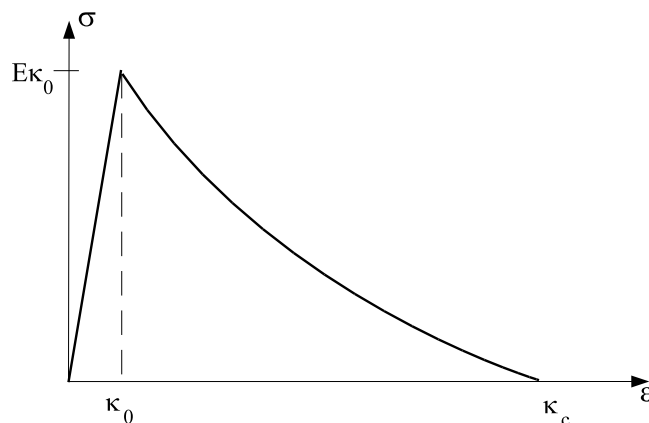
$$d = 1 - \frac{\kappa_i \kappa_c - \kappa}{\kappa \kappa_c - \kappa_i}, \quad (3.78)$$



Rysunek 3.5: Liniowe osłabienie

- *Oslabienie potęgowe.* Ewolucja zmiennej zniszczenia, w przypadku osłabienia potęgowego, wyrażona jest wzorem (Rys. 3.6)

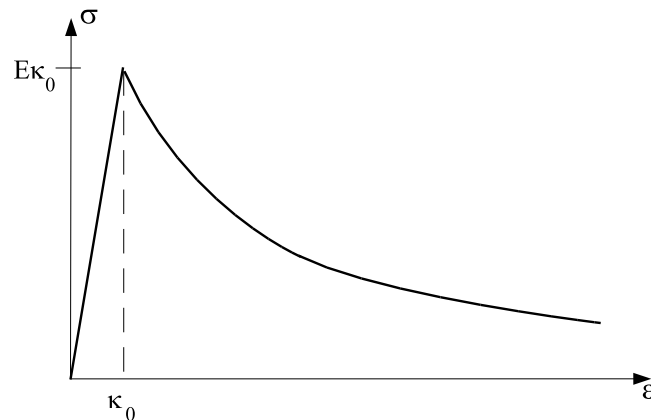
$$d = 1 - \left(\frac{\kappa_i}{\kappa}\right)^\alpha \left(\frac{\kappa_c - \kappa}{\kappa_c - \kappa_i}\right)^\beta, \quad (3.79)$$



Rysunek 3.6: Osłabienie potęgowe

- *Oslabienie wykładnicze* Osłabienie wykładnicze wyraża poniższy wzór.

$$d = 1 - \left(\frac{\kappa_i}{\kappa}\right)^\alpha [1 - \alpha - \alpha e^{-\beta(\kappa - \kappa_i)}], \quad (3.80)$$



Rysunek 3.7: Osłabienie wykładnicze

3.5.3 Model zniszczenia nielokalnego

W zaprezentowanym tu modelu zniszczenia ciągłego (ang. *continuum damage model*) miała zastosowanie zasada lokalnego oddziaływania, sformułowana przez A. Einsteina [1]. Relacja konstytutywna, jak i inne podane powyżej relacje, składające się na model zniszczenia, są zależnościami wielkości w określonym punkcie konstrukcji. Stosując do analizy numerycznej modelu zniszczenia metodę elementów skończonych, dyskretyzujemy konstrukcję pokrywając ją siatką elementów skończonych. Skończenie elementowa dyskretyzacja ma znaczący wpływ, zarówno na rozmieszczenie stref zniszczenia, jak i na wielkość obciążenia niszczącego konstrukcję. Strefy zniszczenia mają tendencję do lokalizowania się w wąskich pasmach. Im drobniejsza dyskretyzacja, tym wcześniej w historii obciążenia obserwuje się powstawanie stref zniszczenia i tym szybciej następuje ich wzrost. W granicy, gdy wielkość elementu dąży do zera, obserwuje się dążenie do zera szerokości stref zniszczenia oraz idealnie kruche zachowanie się materiału, czyli inicjacja zniszczenia jest od razu zniszczeniem konstrukcji, a obciążenie inicjujące zniszczenie jest obciążeniem krytycznym.

By zapobiec tym patologicznym wpływom dyskretyzacji skończenie elementowej zastępujemy nielokalny model zniszczenia, polegający na wprowadzeniu ważonej, uśrednionej wartości odkształceń efektywnych zdefiniowanej następująco

$$\bar{\varepsilon}_{\text{eq}}(\mathbf{x}) = \frac{1}{V_r} \int_V g(\mathbf{r}) \varepsilon_{\text{eq}}(\mathbf{x} + \mathbf{r}) dV(\mathbf{r}), \quad (3.81)$$

gdzie

$$V_r = \int_V g(\mathbf{r}) dV(\mathbf{r}). \quad (3.82)$$

Nielocalne odkształcenie efektywne $\bar{\varepsilon}_{\text{eq}}$ jest zatem pewną średnią wielkością ważoną, obliczaną w oparciu o wartości odkształcenia efektywnego lokalnego, w otoczeniu rozpatrywanego punktu \mathbf{x} , z udziałem wagi g , określającej wpływ otoczenia punktu; zwykle wartość

wag maleje wraz z odległością od punktu \mathbf{x} . Stosowane są różne funkcje wagowe; poniżej przytoczymy dwie definicje. Pierwsza z nich to funkcja Gaussa

$$g(\mathbf{r}) = \exp\left(-\frac{r^2}{l^2}\right), \quad (3.83)$$

gdzie l jest parametrem długości określającym zasięg wpływu wag. Druga funkcja wagowa ma postać:

$$g(\mathbf{r}) = \begin{cases} 1 - \frac{r^2}{R^2} & \text{dla } r \leq R \\ 0 & \text{dla } r > R, \end{cases} \quad (3.84)$$

gdzie podobnie jak poprzednio R jest zasięgiem wpływu zwanym *promieniem oddziaływania*.

3.5.4 Model zniszczenia w sformułowaniu skończenie elementowym

Przedstawione powyżej rozważania zastosowane będą w sformułowaniu MES dla problemu statyki. Użyty zostanie model konstytutywny Hooke'a uzupełniony o model zniszczenia (3.65). Wprowadzimy dodatkowe oznaczenia

$$\mathbf{C}_D = (1 - d)\mathbf{C}, \quad (3.85)$$

zatem

$$\boldsymbol{\sigma} = \mathbf{C}_D \boldsymbol{\varepsilon}, \quad (3.86)$$

Konsystentny moduł styczny wyznaczamy następująco

$$\mathbf{C}_D^* = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} = (1 - d)\mathbf{C} - \frac{\partial d}{\partial \boldsymbol{\varepsilon}} \mathbf{C} \boldsymbol{\varepsilon}, \quad (3.87)$$

gdzie pochodna zmiennej zniszczenia wyraża się wzorem

$$\frac{\partial d}{\partial \boldsymbol{\varepsilon}} = \frac{dk}{d\varepsilon_{\text{eq}}} \frac{d\varepsilon_{\text{eq}}}{d\boldsymbol{\varepsilon}} \quad (3.88)$$

Stosując zasadę prac wirtualnych

$$\int_{\Omega} \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\Omega = \int_{\Omega} \delta u^T \mathbf{p} d\Omega + \int_{\partial\Omega_{\sigma}} \delta u^T \mathbf{t} d\partial\Omega_{\sigma}, \quad (3.89)$$

oraz model konstytutywny, z uwzględnieniem zniszczenia, (3.86) otrzymujemy

$$\delta \mathbf{u}^T \int_{\Omega} \mathbf{B}^T \mathbf{C}_D \mathbf{B} d\Omega \mathbf{q} = \int_{\Omega} \delta \mathbf{u}^T \mathbf{N}^T \mathbf{p} d\Omega + \int_{\partial\Omega_{\sigma}} \delta \mathbf{u}^T \mathbf{N}^T \mathbf{t} d\partial\Omega_{\sigma}. \quad (3.90)$$

Wprowadzając poniższe oznaczenie dla macierzy sieciowej i wektora obciążenia

$$\mathbf{K}_S = \int_{\Omega} \mathbf{B}^T \mathbf{C}_D \mathbf{B} d\Omega, \quad \mathbf{P} = \int_{\Omega} \mathbf{N}^T \mathbf{p} d\Omega + \int_{\partial\Omega_{\sigma}} \mathbf{N}^T \mathbf{t} d\partial\Omega_{\sigma}. \quad (3.91)$$

Równanie to można rozwiązywać iteracyjnie, stosując następujący schemat iteracyjny

$$\mathbf{K}_S(\mathbf{q}_i)\mathbf{q}_{i+1} = \mathbf{P}. \quad (3.92)$$

Zbieżność tego procesu iteracyjnego jest zaledwie liniowa, więc warto zastosować sformułowanie przyrostowe

$${}^{i+1}\boldsymbol{\sigma} = {}^i\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}, \quad (3.93)$$

$$\Delta\boldsymbol{\sigma} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \Delta\boldsymbol{\varepsilon} = {}^i\mathbf{C}_T^* \mathbf{B} \Delta\mathbf{q}. \quad (3.94)$$

Podstawiając do zasady prac wirtualnych dla ${}^{i+1}\boldsymbol{\sigma}$ otrzymujemy

$$\delta\mathbf{u}^T \int_{\Omega} \mathbf{B}^{Ti} \mathbf{C}_D \mathbf{B} d\Omega {}^i\mathbf{q} + \delta\mathbf{u}^T \int_{\Omega} \mathbf{B}^T \mathbf{C}_D^* \mathbf{B} d\Omega \Delta\mathbf{q} = \quad (3.95)$$

$$= \int_{\Omega} \delta\mathbf{u}^T \mathbf{N}^T \mathbf{p} d\Omega + \int_{\partial\Omega_\sigma} \delta\mathbf{u}^T \mathbf{N}^T \mathbf{t} d\partial\Omega_\sigma \quad (3.96)$$

co daje równanie przyrostowe

$$\mathbf{K}_T \Delta\mathbf{q} = {}^{i+1}\mathbf{P} - {}^i\mathbf{F}, \quad (3.97)$$

gdzie

$$\mathbf{K}_T = \int_{\Omega} \mathbf{B}^T \mathbf{C}_D^* \mathbf{B} d\Omega, \quad (3.98)$$

$${}^{i+1}\mathbf{F} = \int_{\Omega} \mathbf{B}^T \mathbf{C}_D \mathbf{B} {}^i\mathbf{q} d\Omega. \quad (3.99)$$

Równanie (3.97) można rozwiązywać metodą Newtona-Raphsona, zapewniającą kwadratową zbieżność.

3.5.5 Termolepkosprężystość ze zniszczeniem

Opisany w trzecim rozdziale model termolepkosprężysty uzupełniony zostanie obecnie o model zniszczenia. Równanie konstytutywne będzie zatem miało postać

$$\sigma_{ij}(\tau) = (1-d) \int_0^\tau C_{ijkl}(\xi - \xi') \frac{d}{d\tau'} [\varepsilon_{kl}(\tau') - \varepsilon_{kl}^{(\theta)}(\tau')] d\tau'. \quad (3.100)$$

Przekształcając, podobnie jak w rozdziale trzecim, mamy

$$\begin{aligned} \sigma_{ij}(\tau) &= (1-d) \int_0^\tau \left[C_{ijkl}^\infty + \tilde{C}_{ijkl} \exp\left[-\frac{\xi - \xi'}{\varrho}\right] \right] \frac{d}{dt'} [\varepsilon_{ij}(\tau') - \varepsilon_{ij}^{(\theta)}(\tau')] d\tau' = \\ &= (1-d) \int_0^\tau \left[C_{ijkl}^0 - \tilde{C}_{ijkl} \left[1 - \exp\left[-\frac{\xi - \xi'}{\varrho}\right] \right] \right] \frac{d}{dt'} [\varepsilon_{ij}(\tau') - \varepsilon_{ij}^{(\theta)}(\tau')] d\tau' = \\ &= (1-d) \left[C_{ijkl}^0 [\varepsilon_{kl}(\tau) - \varepsilon_{kl}^{(\theta)}(\tau)] - \sigma_{ij}^{(c)}(\tau) \right]. \end{aligned} \quad (3.101)$$

Przyrostowa forma równania konstytutywnego ma zatem postać

$${}^{t+\Delta t}\boldsymbol{\sigma} = (1 - {}^{t+\Delta t}d) [\mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}]. \quad (3.102)$$

Zmienna zniszczenia ${}^{t+\Delta t}d$ oznacza zależność od odkształceń na kroku $t+\Delta t$ czyli $d({}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}})$. Stosując podobne przekształcenia, jak w drugim rozdziale mamy

$$\begin{aligned} {}^{t+\Delta t}\boldsymbol{\sigma} &= (1 - {}^{t+\Delta t}d) \left[\mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - \left[\frac{1}{\Delta t} + \frac{1}{{}^{t+\Delta t}A\varrho} \right]^{-1} \left[\frac{{}^t\boldsymbol{\sigma}^{(c)}}{\Delta t} + \frac{\tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{{}^{t+\Delta t}A\varrho} \right] \right] = \\ &= (1 - {}^{t+\Delta t}d) \left[{}^{t+\Delta t}\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - \frac{{}^{t+\Delta t}A\varrho}{\Delta t + {}^{t+\Delta t}A\varrho} {}^t\boldsymbol{\sigma}^{(c)} \right], \end{aligned} \quad (3.103)$$

gdzie $\bar{\mathbf{C}}^*$ nie jest tym razem modulem konsystentnym, a jedynie oznaczeniem wprowadzonym dla przejrzystości i analogii z oznaczeniami w rozdziale trzecim

$$\begin{aligned} {}^{t+\Delta t}\bar{\mathbf{C}}^* &= \mathbf{C}^0 - \left[\frac{1}{\Delta t} + \frac{1}{{}^{t+\Delta t}A\varrho} \right]^{-1} \frac{\tilde{\mathbf{C}}}{{}^{t+\Delta t}A\varrho} = \\ &= \mathbf{C}^0 - \frac{\Delta t \tilde{\mathbf{C}}}{\Delta t + {}^{t+\Delta t}A\varrho}. \end{aligned} \quad (3.104)$$

Podstawiając do równania prac wirtualnych (3.38) i stosując skończenie elementową aproksymację (3.39) oraz uwzględniając oznaczenia (3.41), mamy

$$\begin{aligned} &({}^t\mathbf{q} + \Delta\mathbf{q})^T \left[\int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \mathbf{C}^0 \mathbf{B} d\Omega \right] \delta\mathbf{q} - \left[\int_{\Omega} {}^{t+\Delta t}D^{(c)t} (1 - {}^{t+\Delta t}d) \boldsymbol{\sigma}^{(c)T} \mathbf{B} d\Omega \right] \delta\mathbf{q} + \\ &- ({}^t\mathbf{q} + \Delta\mathbf{q})^T \left[\int_{\Omega} (1 - {}^{t+\Delta t}d) \mathbf{B}^T {}^{t+\Delta t}\tilde{D}\tilde{\mathbf{C}}\mathbf{B} d\Omega \right] \delta\mathbf{q} - \\ &\left[\int_{\Omega} (1 - {}^{t+\Delta t}d) \boldsymbol{\alpha}^T \theta_0 {}^{t+\Delta t}\mathbf{C}^{*T} \mathbf{B} d\Omega \right] \delta\mathbf{q} + \\ &\boldsymbol{\theta}^T \left[\int_{\Omega} (1 - {}^{t+\Delta t}d) \bar{\mathbf{N}}^T \boldsymbol{\alpha}^T {}^{t+\Delta t}\mathbf{C}^{*T} \mathbf{B} d\Omega \right] \delta\mathbf{q} = \left[\int_{\Omega} \rho ({}^t\mathbf{f} + \Delta\mathbf{f})^T \mathbf{N} d\Omega \right] \delta\mathbf{q} + \\ &\left[\int_{\partial\Omega_{\sigma}} ({}^t\mathbf{t} + \Delta\mathbf{t})^T \mathbf{N} d\partial\Omega_{\sigma} \right] \delta\mathbf{q}. \end{aligned} \quad (3.105)$$

Wprowadzając analogiczne do (3.41)-(3.53)

$$\mathbf{K}^0 = \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \mathbf{C}^0 \mathbf{B} d\Omega, \quad (3.106)$$

$$\tilde{\mathbf{K}} = \left[\int_{\Omega} \tilde{D} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \tilde{\mathbf{C}} \mathbf{B} d\Omega \right], \quad (3.107)$$

$$\mathbf{K}^* = \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \mathbf{C}^* \mathbf{B} d\Omega = \mathbf{K}^0 - \tilde{\mathbf{K}}, \quad (3.108)$$

$$\mathbf{K}_S = \int_{\Omega} \bar{\mathbf{N}}^T \boldsymbol{\alpha}^T (1 - {}^{t+\Delta t}d) \mathbf{C}^* \mathbf{B} d\Omega, \quad (3.109)$$

$${}^{t+\Delta t}\mathbf{Q} = \left[\int_{\Omega} (\hat{\mathbf{t}}\mathbf{f} + \Delta\hat{\mathbf{t}})^T \mathbf{N} d\Omega \right] + \left[\int_{\partial\Omega_{\sigma}} (\hat{\mathbf{t}} + \Delta\hat{\mathbf{t}})^T \mathbf{N} d\partial\Omega_{\sigma} \right], \quad (3.110)$$

$$\mathbf{F}^0 = \left[\int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \mathbf{C}^0 \mathbf{B} d\Omega \right] {}^t\mathbf{q}, \quad (3.111)$$

$$\mathbf{F}^{(\theta)} = \int_{\Omega} (1 - {}^{t+\Delta t}d) \boldsymbol{\alpha}^T \theta_0 \mathbf{C}^* \mathbf{B} d\Omega, \quad (3.112)$$

$$\tilde{\mathbf{F}} = \left[\int_{\Omega} \tilde{D} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \tilde{\mathbf{C}} \mathbf{B} d\Omega \right] {}^t\mathbf{q} = \tilde{\mathbf{K}} {}^t\mathbf{q}, \quad (3.113)$$

$$\tilde{\mathbf{F}}_D = \left[\int_{\Omega} \tilde{D} \mathbf{B}^T (1 - {}^{t+\Delta t}d) \tilde{\mathbf{C}} \mathbf{B} d\Omega \right] \Delta\mathbf{q} = \tilde{\mathbf{K}} \Delta\mathbf{q}, \quad (3.114)$$

$$\bar{\mathbf{F}}^{(c)} = \left[\int_{\Omega} D^{(c)} \mathbf{B}^T (1 - {}^{t+\Delta t}d) {}^t\boldsymbol{\sigma}^{(c)} d\Omega \right], \quad (3.115)$$

$$\mathbf{F}^{(c)} = \tilde{\mathbf{F}} + \bar{\mathbf{F}}^{(c)}, \quad (3.116)$$

$${}^{t+\Delta t}\hat{\mathbf{Q}} = {}^{t+\Delta t}\mathbf{Q} - \mathbf{F}^0 + \mathbf{F}^{(c)} + \mathbf{F}^{(\theta)}. \quad (3.117)$$

Używając powyższych oznaczeń można równanie (3.95) napisać w postaci

$$\left[\mathbf{K}^0 - \tilde{\mathbf{K}} \right] \Delta\mathbf{q} = {}^{t+\Delta t}\mathbf{Q} - \mathbf{F}^0 + \mathbf{F}^{(c)} + \mathbf{F}^{(\theta)}. \quad (3.118)$$

Wszystkie składowe powyższego równania, oprócz obciążenia ${}^{t+\Delta t}\mathbf{Q}$, zależne są poprzez zmienną zniszczenia od przemieszczeń, zatem do rozwiązania tego równania zastosować należy proces iteracyjny

$$\left[{}^i\mathbf{K}^0 - {}^i\tilde{\mathbf{K}} \right] {}^{i+1}\Delta\mathbf{q} = {}^{t+\Delta t}\mathbf{Q} - {}^i\mathbf{F}^0 + {}^i\mathbf{F}^{(c)} + {}^i\mathbf{F}^{(\theta)}. \quad (3.119)$$

Powyższy proces iteracyjny jest tylko liniowo zbieżny zatem lepiej zbudować równanie MES dla lepkosprężystości, z zastosowaniem konsystentnej macierzy stycznej, a do rozwiązania metodą Newtona-Raphsona.

Konsystentny moduł styczny wyznaczony zostanie w następujący sposób

$${}^{t+\Delta t}\mathbf{C}^* = \frac{\partial {}^{t+\Delta t}\boldsymbol{\sigma}}{\partial {}^{t+\Delta t}\boldsymbol{\varepsilon}} = \quad (3.120)$$

$$= (1 - {}^{t+\Delta t}d)\bar{\mathbf{C}}^* - \frac{\partial d}{\partial {}^{t+\Delta t}\boldsymbol{\varepsilon}} \left[{}^{t+\Delta t}\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - \frac{{}^{t+\Delta t}A\varrho}{{}^{t+\Delta t}A} {}^t\boldsymbol{\sigma}^{(c)} \right]. \quad (3.121)$$

Podstawiając do równań (3.93), (3.94) otrzymujemy

$$\begin{aligned} {}^{t+\Delta t}\boldsymbol{\sigma}^{i+1} &= (1 - {}^{t+\Delta t}d) \left[\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}^{(i)} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right] + (1 - {}^{t+\Delta t}d)\bar{\mathbf{C}}^* \Delta\boldsymbol{\varepsilon} \\ &\quad - \frac{\partial {}^{t+\Delta t}d}{\partial {}^{t+\Delta t}\boldsymbol{\varepsilon}^{(i)}} \left[\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right] \Delta\boldsymbol{\varepsilon}. \end{aligned} \quad (3.122)$$

Zatem po podstawieniu do równania prac wirtualnych mamy

$$\begin{aligned} &\delta\mathbf{q}^T \int_{\Omega} \mathbf{B}^T \left[(1 - {}^{t+\Delta t}d^i) \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}^{(i)} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] d\Omega + \\ &+ \delta\mathbf{q}^T \int_{\Omega} \mathbf{B}^T \left[(1 - {}^{t+\Delta t}d^i)\bar{\mathbf{C}}^* - \frac{\partial {}^{t+\Delta t}d^i}{\partial \boldsymbol{\varepsilon}} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}^{(i)} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B}\Delta\mathbf{q} d\Omega = \\ &= \int_{\Omega} \delta\mathbf{u}^T \mathbf{N}^T \mathbf{p} d\Omega + \int_{\partial\Omega_{\sigma}} \delta\mathbf{u}^T \mathbf{N}^T \mathbf{t} d\partial\Omega_{\sigma}. \end{aligned} \quad (3.123)$$

Oznaczamy poszczególne wyrazy równania

$${}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}^{(i)} = \mathbf{B}\mathbf{q}^{(i)} - \boldsymbol{\alpha}\mathbf{N}\boldsymbol{\theta} + \boldsymbol{\alpha}\theta_0, \quad (3.124)$$

$$\mathbf{K}_T = \int_{\Omega} \mathbf{B}^T \left[(1 - {}^{t+\Delta t}d^i)\bar{\mathbf{C}}^* - \frac{\partial {}^{t+\Delta t}d^i}{\partial \boldsymbol{\varepsilon}} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}^{(i)} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} d\Omega, \quad (3.125)$$

$$\mathbf{K}_s = - \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i)\boldsymbol{\alpha}\mathbf{N} d\Omega, \quad (3.126)$$

$$\mathbf{F}^{(i)} = \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i)\bar{\mathbf{C}}^* \mathbf{B} {}^{t+\Delta t}\mathbf{q}^{(i)} d\Omega, \quad (3.127)$$

$$\mathbf{F}_c^{(i)} = \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} d\Omega, \quad (3.128)$$

$$\mathbf{F}_{\theta}^{(i)} = \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i)\bar{\mathbf{C}}^* \boldsymbol{\alpha}\theta_0 d\Omega, \quad (3.129)$$

$${}^{t+\Delta t}\mathbf{Q} = \int_{\Omega} (\hat{\mathbf{f}} + \Delta\hat{\mathbf{f}})^T \mathbf{N} d\Omega + \int_{\partial\Omega_{\sigma}} (\hat{\mathbf{t}} + \Delta\hat{\mathbf{t}})^T \mathbf{N} d\partial\Omega_{\sigma}, \quad (3.130)$$

$$\hat{\mathbf{Q}} = {}^{t+\Delta t}\mathbf{Q} - \mathbf{F}^{(i)} - \mathbf{F}_c^{(i)} - \mathbf{F}_{\theta}^{(i)}. \quad (3.131)$$

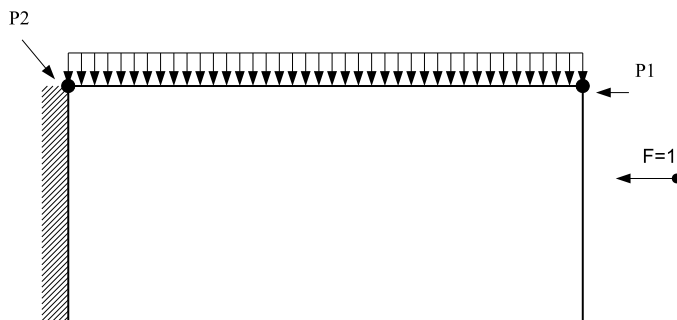
Mając komplet oznaczeń, możemy napisać układ równań dla termolepkosprężystego materiału ze zniszczeniem, z częścią mechaniczną w sformułowaniu przyrostowym

$$\begin{bmatrix} {}^{t+\alpha\Delta t}\widehat{\mathbf{K}} & 0 \\ \mathbf{K}_S & \mathbf{K}_T \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t}\boldsymbol{\theta} \\ \Delta\mathbf{q} \end{bmatrix} = \begin{bmatrix} {}^{t+\alpha\Delta t}\widehat{\mathbf{P}} \\ \widehat{\mathbf{Q}} \end{bmatrix}. \quad (3.132)$$

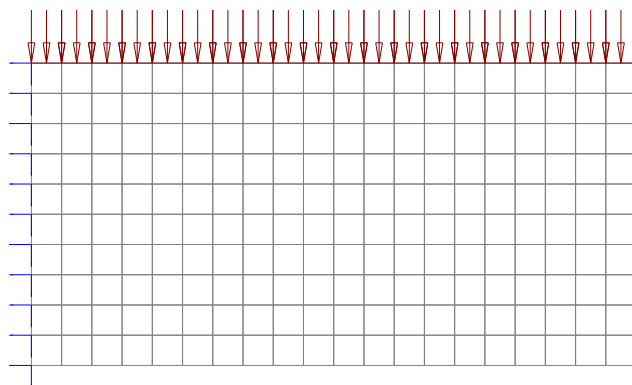
Powyższy układ równań składa się z dwóch podukładów, które można rozwiązać oddzielnie. Pierwsze równanie niestacjonarnego przepływu ciepła może być rozwiązane w pierwszej kolejności, a następnie po podstawieniu pola temperatury do układu, rozwiązane zostanie drugie równanie postaci

$$\mathbf{K}_T\Delta\mathbf{q} = \widehat{\mathbf{Q}} - \mathbf{K}_S {}^{t+\Delta t}\boldsymbol{\theta}. \quad (3.133)$$

3.6 Przykład numeryczny - zniszczenie wspornika lepkosprężystego



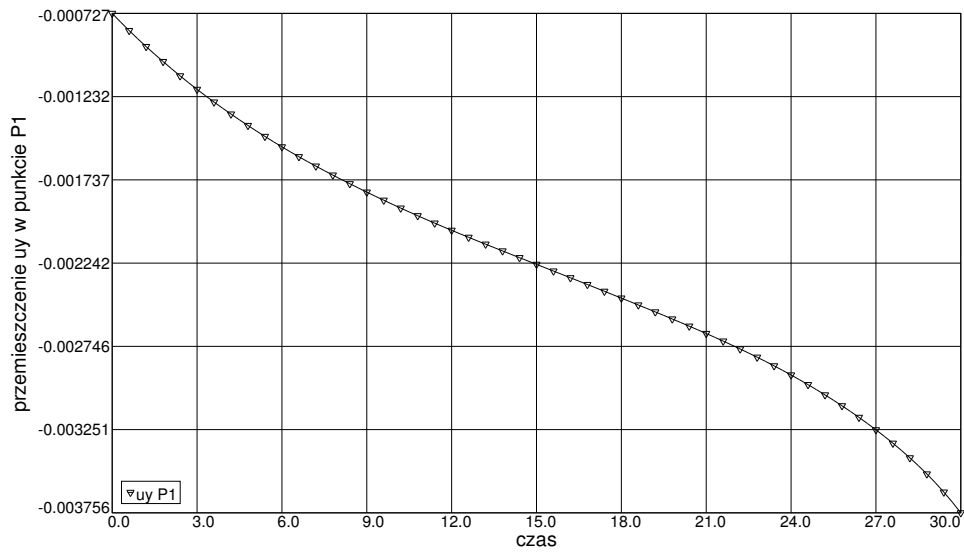
Rysunek 3.8: Schemat wspornika



Rysunek 3.9: Dyskretyzacja MES wspornika

Schemat statyczny analizowanego wspornika przedstawiony jest na Rys. 3.8. Dane zadania są następujące: materiał termolepkosprężysty, $\tau_0 = 30s.$, 100 równych kroków czasowych zastosowano w tym teście. Użyto następujących wartości stałych materiałowych $E^0 = 500$, $E^\infty = 100$, $\varrho = 1$, $\nu = 0.3$, $\kappa = \frac{\lambda}{\rho c} = 1$, wartość strumienia ciepła na swobodnym końcu pręta wspornika $\hat{q} = 1$. Dane dotyczące modelu zniszczenia: $\alpha_1 = 0.96$ $\alpha_2 = 350$. Stosunek wytrzymałości na ściskanie do wytrzymałości na rozciąganie $k_{ct} = 10$ $k_i = 1.0e - 4$. Promień wpływu dla nielokalnego zniszczenia $R = 0.2$. Na utwierdzonym końcu pręta zadano warunki brzegowe dla temperatury $T = 0$. Funkcja zależności termicznej $A(\theta)$ z równania Eq. 3.18 przyjmuje wartość $A(\theta) = 0.5 \cdot T + 2.5$. Dyskretyzacja skończenie-elementowa zawiera 200 czterowzłowych, liniowych elementów typu Lagrange w siatce 20x10 elementów. Równomierne obciążenie ciągłe ma wartość $q=2.5$. Wspornik poddano analizie pełzania termolepkosprężystego ze zniszczeniem nielokalnym. Wykres przemieszczenia górnego końca wspornika (P1) zawiera Rys. 3.10. Rys. 3.11 zawiera mapę warstwicową rozkładu stref zniszczenia po upływie rozpatrywanego czasu analizy. Zniszczenie następuje w strefie rozciąganej zgodnie z cechami ośrodka. Przedstawiono również

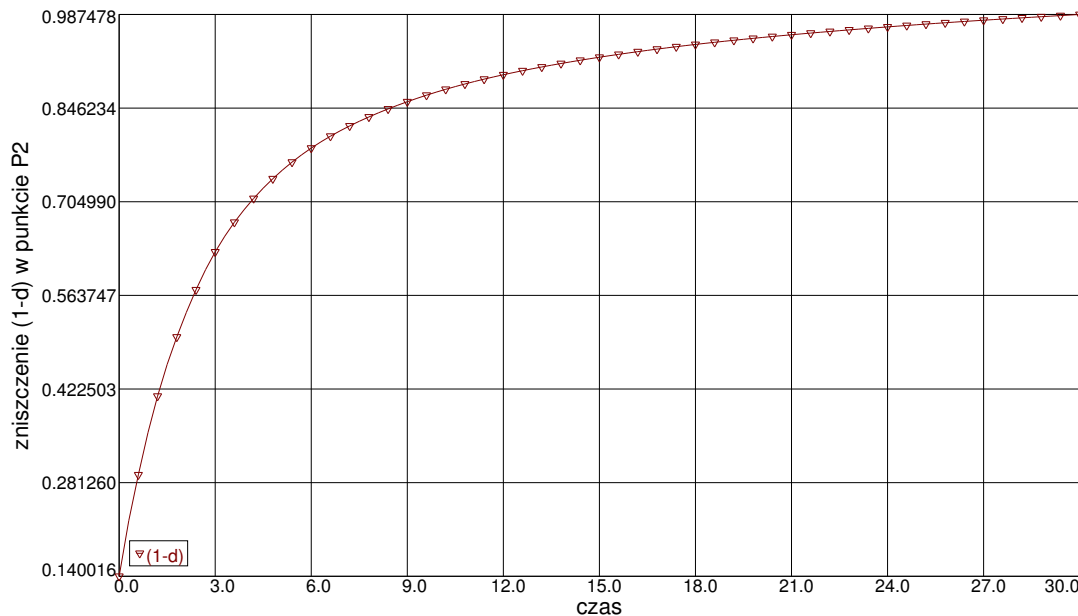
na rysunku Rys. 3.12 wykres zmian funkcji współczynnika zniszcenia ($1 - d$) punkcie P2 w którym rozwija się zniszczenie.



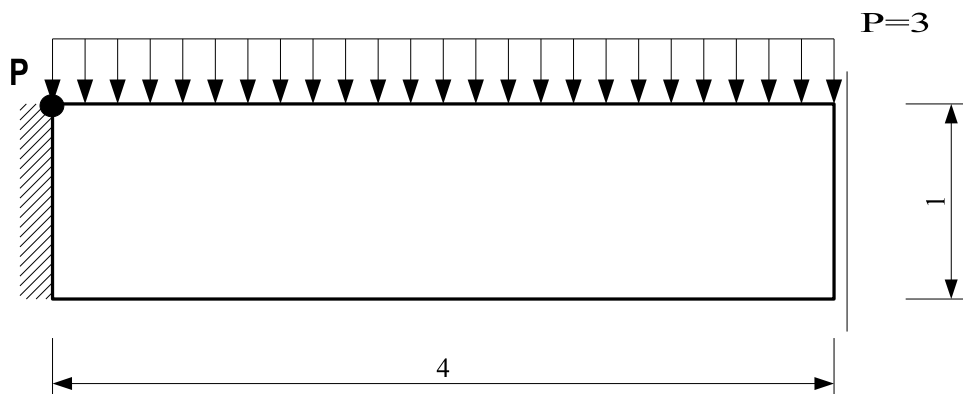
Rysunek 3.10: Pełzanie górnego, swobodnego wierzchołka pręta wspornika



Rysunek 3.11: Rozkład stref zniszczenia konstrukcji wspornika

Rysunek 3.12: Wykres zmian w czasie wartości $(1 - d)$

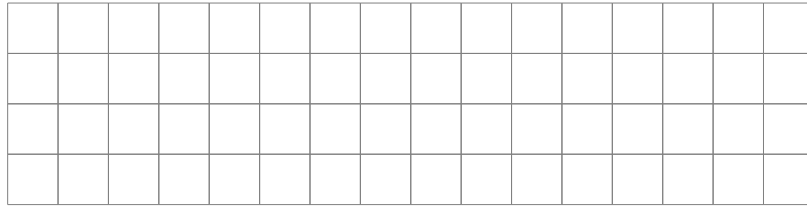
3.7 Przykład numeryczny - lokalne i nielocalne zniszczenie belki lepkosprężystej.



Rysunek 3.13: Schemat statyczny belki

W przykładzie tym przedmiotem analizy będzie belka utwierdzona z lepkosprężystym materiałem z uwzględnieniem zniszczenia. Zadanie rozwiązane zostanie wielokrotnie, dla różnych dyskretyzacji MES i przy założeniu lokalnego oraz nielokalnego sformułowania zniszczenia. Schemat statyczny belki uwidoczniiony jest na Rys. 3.13. Model uwzględnia warunki symetrii belki. W zadaniu tym nie uwzględniono wpływu temperatury gdyż nie ma ona istotnego wpływu na pokazanie różnic pomiędzy lokalnym a nielokalnym modelem zniszczenia. Dane zadania są następujące: analizowany przedział czasowy $\tau = 0 - 30s.$, 100 równych kroków czasowych zastosowano w tym teście. Stałe materiałowe lepkosprężystości są następujące: $E^0 = 210000$, $E^\infty = 80000$, $\rho = 1$, $\nu = 0.166667$, Dane mo-

delu zniszczenia są następujące $\alpha_1 = 0.96$ $\alpha_2 = 350$. Stosunek wytrzymałości na ściskanie do wytrzymałości na rozciąganie $k_{ct} = 10$ $k_i = 1.0e - 4$. Promień wpływu dla nielokalnego zniszczenia $R = 0.5$. Na rysunkach 3.14-3.16 przedstawiono przykłady porównawcze dla różnych wielkości siatki elementów czterowęzłowych Lagrange. W ujęciu lokalnym, wraz z rosnącą liczbą stopni swobody nie obserwuje się zwiększania się dokładności rozwiązania. Strefy zniszczenia mają tendencję do lokalizowania się wąskich pasmach (3.14b,3.15b,3.16b). Takie samo zjawisko obserwuje się stosując siatkę trójkątów o różnej długości krawędzi. Przykład belki rozwiązano używając trzech różnych siatek trójkątów o długości krawędzi kolejno 0.3, 0.2, 0.1. Rezultaty obliczeń zamieszczono na rysunkach (3.20,3.21,3.22). Lokalizacja stref zniszczenia wraz ze zmniejszaniem rozmiaru elementów trójkątnych widoczna jest na rysunkach (3.20b,3.21b,3.22b). Wyraźną poprawę obiektywności rezultatów widać w rozwiązaniach, w których stosowano ujęcie nielokalne. Niezależnie od zastosowanej siatki, strefy zniszczenia we wszystkich wykonanych tu rozwiązaniach z użyciem nielokalnego zniszczenia wyglądają niemal jednakowo (3.14c,3.15c,3.16c,3.19c,(3.20c,3.21c,3.22c). Dla zadań z siatką elementów czterowęzłowych typu Lagrange użyto techniki adaptacji siatki regularnej zarówno w ujęciu lokalnym jak i nielokalnym. Poprawianie siatki zadania zniszczenia lokalnego doprowadziło do otrzymania rezultatów z wyraźnie zlokalizowanymi strefami zniszczenia. Rys. 3.17. Strefy zniszczenia dla siatki adaptacyjnej modelu nielokalnego Rys. 3.18 są takie same jak w innych rozwiązywanych w tym przykładzie zadaniach ze zniszczeniem nielokalnym. Na wszystkich zamieszczonych w tym przykładzie mapach warstwicowych, obrazujących rozkład stref zniszczenia kolor niebieski oznacza materiał w stanie pierwotnym $d = 0$, natomiast kolor czerwony oznacza strefę niezerowego współczynnika zniszczenia d z maksymalną wartością w punkcie P równą $d = 0.99$.



a)

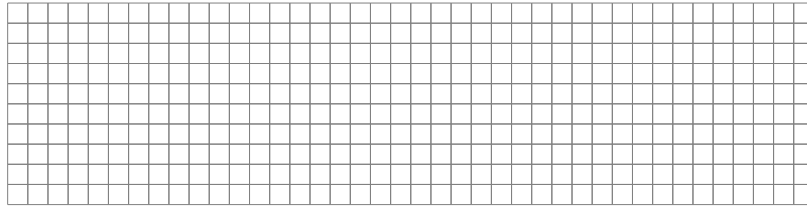


b)



c)

Rysunek 3.14: Porównanie lokalnego i nielokalnego zniszczenia w belce modelowanej elementami z czterowęzłową aproksymacją Lagrange: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym



a)

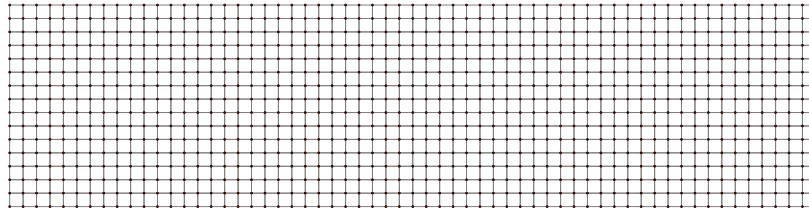


b)



c)

Rysunek 3.15: Porównanie lokalnego i nielokalnego zniszczenia w belce modelowanej elementami z czterowęzłową aproksymacją Lagrange: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym



a)

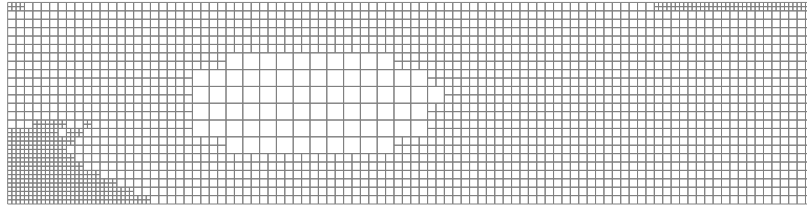


b)



c)

Rysunek 3.16: Porównanie lokalnego i nielocalnego zniszczenia w belce modelowanej elementami z czterowęzłową aproksymacją Lagrange: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym

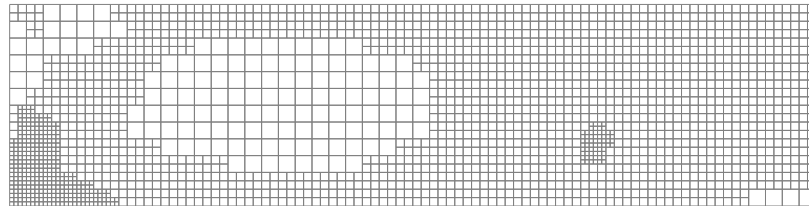


a)



b)

Rysunek 3.17: Zastosowanie techniki adaptacyjnej dla siatek regularnych w lokalnym modelu zniszczenia: a) siatka adaptacyjna MES, b) zniszczenie w sformułowaniu lokalnym

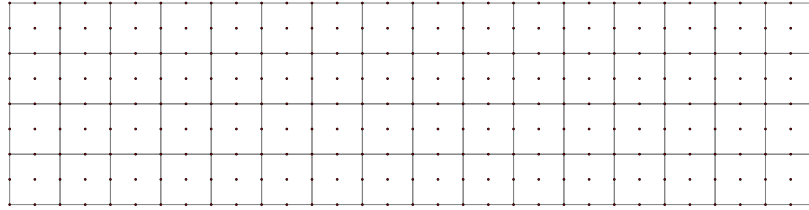


a)



b)

Rysunek 3.18: Zastosowanie techniki adaptacyjnej dla siatek regularnych w nielokalnym modelu zniszczenia: a) siatka adaptacyjna MES, b) zniszczenie w sformułowaniu nielokalnym



a)

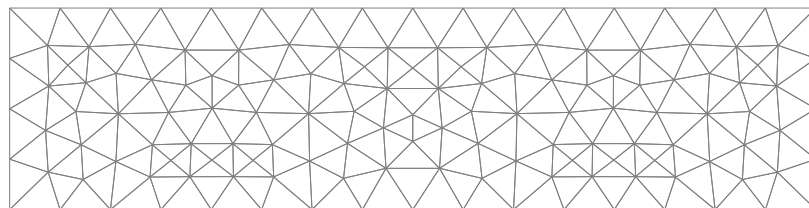


b)

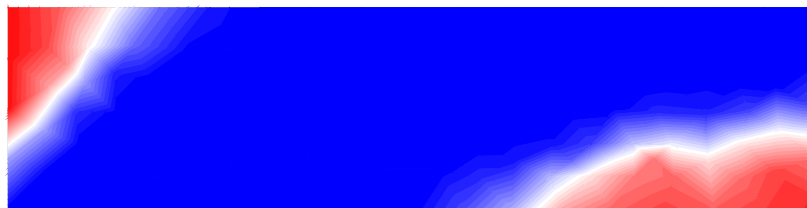


c)

Rysunek 3.19: Porównanie lokalnego i nielokalnego zniszczenia w belce modelowanej elementami z dziewięciowęzłową aproksymacją Lagrange: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym



a)

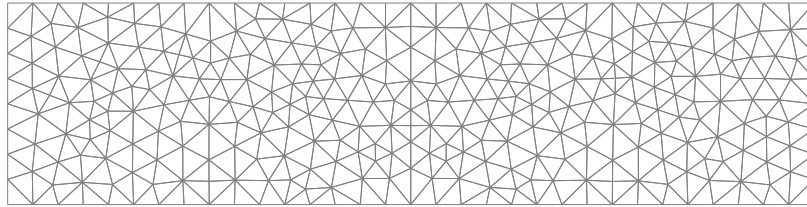


b)



c)

Rysunek 3.20: Porównanie lokalnego i nielokalnego zniszczenia w belce modelowanej elementami trójkątnymi wielkością krawędzi 0.3: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym



a)

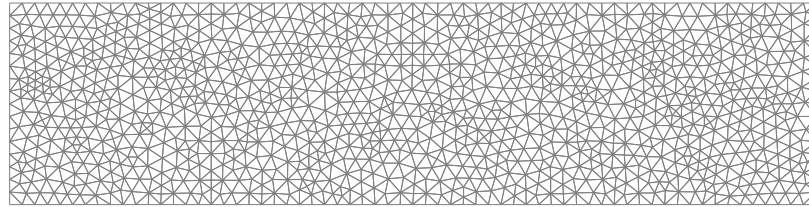


b)



c)

Rysunek 3.21: Porównanie lokalnego i nielokalnego zniszczenia w belce modelowanej elementami trójkątnymi wielkością krawędzi 0.2: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym



a)



b)



c)

Rysunek 3.22: Porównanie lokalnego i nielokalnego zniszczenia w belce modelowanej elementami trójkątnymi wielkością krawędzi 0.1: a) siatka MES, b) zniszczenie w sformułowaniu lokalnym, c) zniszczenie w sformułowaniu nielokalnym

Analiza wrażliwości dla zagadnień termolepkosprężystych

4.1 Podstawowe pojęcia

Najbardziej efektywnymi metodami zarówno optymalizacji jak i niezawodności wydają się być obecnie metody gradientowe. Dlatego istotnym zagadnieniem problemu optymalizacji jest możliwość obliczania gradientów funkcji celu oraz gradientu ograniczeń ze względu na zmienne projektowe. Wyznaczaniem tych gradientów zajmuje się *analiza wrażliwości*.

Funkcjami celu dla optymalizacji są zwykle rezultaty analizy skończenie elementowej takie jak np. naprężenia czy przemieszczenia lub ich funkcje. Zmiennymi projektowymi zaś dowolne parametry charakteryzujące zadanie takie jak np. stałe materiałowe czy parametry kształtu. Przykładowe równanie MES dla problemu zależnego od czasu dla zadanego kroku czasowego wygląda następująco:

$$\mathbf{K}^*(h)\Delta\mathbf{q}(h) = {}^t\mathbf{Q}(h), \quad (4.1)$$

gdzie \mathbf{K}^* to macierz konsystentna, \mathbf{Q} wektor prawych stron, a $\Delta\mathbf{q}(h)$ wektor przyrostów przemieszczeń węzłowych na kroku czasowym. Macierz konsystentna oraz wektor prawych stron jawnie zależą od parametru projektowego, natomiast wektor przyrostów przemieszczeń niejawnie, jako rozwiązanie równania

$$\Delta\mathbf{q}(h) = \mathbf{K}^*(h)^{-1} {}^t\mathbf{Q}(h). \quad (4.2)$$

Istnieje kilka metod, przy pomocy których możemy obliczać wrażliwość. Poniżej omówione zostaną trzy z nich.

4.1.1 Metody obliczania wrażliwości

4.1.1.1 Metoda różnic skończonych

Metoda różnic skończonych polega na wyznaczeniu wrażliwości z wyrażenia przyrostowego. Przykładowo

$$\frac{\Delta \mathbf{q}}{\Delta h} = \frac{\mathbf{q}(h + \Delta h) - \mathbf{q}(h)}{\Delta h}. \quad (4.3)$$

Wrażliwość żądanej wielkości na parametr projektowy wyznacza się obliczając dwukrotnie wartość funkcji, raz dla parametru projektowego h , a następnie dla $h + \Delta h$ i podstawienie otrzymanych wartości do wzoru (4.3). Zaletą metody (a właściwie powodem, dla którego w ogóle jest używana) jest łatwość implementacji. W prosty sposób można używać istniejących procedur, czy nawet komercyjny kod, co rozwiązuje praktycznie dowolne problemy. Innym powodem stosowania metody różnic skończonych jest niemożność zastosowania innych metod, ze względu na dużą złożoność problemu (np. bardzo skomplikowane prawa konstytutywne), dla którego zastosowanie innych metod jest niemożliwe. W tworzonym w ramach tej pracy systemie metoda różnic skończonych używana jest jako funkcja diagnostyczna do sprawdzania poprawności kodu wrażliwości obliczonej metodą DDM.

4.1.1.2 Metoda bezpośredniego różniczkowania

Metodę tę oznacza się symbolem DDM od angielskiej nazwy *direct differentiation method*. W tej metodzie różniczkujemy bezpośrednio równanie problemu (4.1), otrzymując, po przekształceniu, następujące równanie

$$\mathbf{K}^* \frac{d\Delta \mathbf{q}}{dh} = \frac{d^t \mathbf{Q}}{dh} - \frac{d\mathbf{K}^*}{dh} \Delta \mathbf{q}. \quad (4.4)$$

Do wyznaczenia wrażliwości potrzebne będzie rozwiązanie problemu równowagi, czyli wektor przyrostów $\Delta \mathbf{q}$. Analizując równanie 4.4 widać, że obliczenie wrażliwości nie jest czasochłonne. Jeśli do rozwiązywania układów równań liniowych wykorzystuje się metody dekompozycyjne, to do obliczenia wrażliwości można wykorzystać zdekomponowaną macierz sztywności w procesie rozwiązywania problemu równowagi. Zatem koszt obliczenia wrażliwości to tylko wyznaczenie niewiadomych w oparciu o wektor prawych stron (podstawienie proste i odwrotne). Do rozwiązania bardzo dużych układów równań liniowych stosuje się metody iteracyjne. W tym przypadku również można obliczać wrażliwość takim kosztem. Stosując np. metodę uogólnionych reziduuw sprzężonych GCR (*Generalized Conjugate Residual*), w której podczas pierwszego rozwiązywania zadania tworzony jest zbiór tzw. wektorów poszukiwań używanych w algorytmie. Wektory te mogą być zastosowane przy drugim i następnym rozwiązaniach dla innych wektorów prawych stron z efektywnością prawie taką, jak podstawienie odwrotne w metodach dekompozycyjnych. Szczegółowe analizy i testy zawarł w swojej książce Lingen [33]. Przykładem nieefektywnej metody w zastosowaniu do analizy wrażliwości metody jest metoda gradientów sprzężonych (CG - *Conjugate Gradient*). W metodzie tej rozwiązanie równania dla drugiego i kolejnych wektorów prawych stron jest tak samo czasochłonne, jak i dla pierwszego, dlatego nie warto jej stosować w analizie wrażliwości. Reasumując: stosując metody iteracyjne do rozwiązywania układów równań liniowych w analizie wrażliwości należy wziąć pod uwagę aspekt efektywności rozwiązania dla wielu wektorów prawych stron.

Algorytm obliczania wrażliwości wygląda następująco:

- rozwiązanie problemu równowagi (4.1).
- obliczenie wektora prawych stron równania (4.4).
- rozwiązanie równania (4.4).

4.1.1.3 Metoda układu sprzężonego

W procesie optymalizacji potrzebna jest nie tylko wrażliwość przemieszczeń, ale też ich funkcji. Funkcja taka ogólnie ma postać:

$$G(\Delta \mathbf{q}(h), h). \quad (4.5)$$

Do wyznaczania wrażliwości takich funkcji pomocna może się okazać metoda układu sprzężonego (ang. adjoint system method ASM). Definiujemy następnie N wymiarowy wektor, zwany wektorem sprzężonym, jako rozwiązanie poniższego układu równań

$$\mathbf{K}^*(h)\boldsymbol{\lambda}(h) = \left(\frac{\partial G}{\partial \Delta \mathbf{q}} \right)^T. \quad (4.6)$$

Wrażliwość funkcji G obliczamy następnie z wyrażenia

$$\frac{dG}{d\Delta \mathbf{q}} = \frac{\partial G}{\partial \Delta \mathbf{q}} + \boldsymbol{\lambda}^T \left(\frac{d^t \mathbf{Q}}{dh} - \frac{dK^*}{dh} \right). \quad (4.7)$$

Główna różnica między tymi metodami jest w efektywności numerycznej. Oznaczając q jako liczbę wektorów prawych stron, k liczbę parametrów projektowych i n liczbę funkcji G , zauważamy, że dla obu metod do rozwiązania zadania trzeba najpierw q razy rozwiązać równanie (4.1), następnie dla DDM należy $q \times k$ rozwiązać równanie (4.4), podczas gdy dla ASM n razy równanie (4.6). Czyli gdy $q \times k < n$, należy wybierać DDM, w przeciwnym wypadku efektywniejsza jest metoda ASM.

4.2 Analiza wrażliwości w problemie przepływu ciepła

Stosując metodę bezpośredniego różniczkowania (DDM), różniczkując (3.6) otrzymujemy równanie wrażliwości dla problemu niestacjonarnego przepływu ciepła

$$\frac{\partial \mathbf{K}}{\partial h} \theta + \mathbf{K} \frac{d\theta}{dh} + \frac{\partial \mathbf{C}}{\partial h} \dot{\theta} + \mathbf{C} \frac{d\dot{\theta}}{dh} = \frac{\partial \mathbf{P}}{\partial h}, \quad (4.8)$$

gdzie $\frac{d\theta(\tau)}{dh}$ jest niewiadomą, stąd wygodnie jest przekształcić to równanie do postaci

$$\mathbf{K} \frac{d\theta}{dh} + \mathbf{C} \frac{d\dot{\theta}}{dh} = \frac{\partial \mathbf{P}}{\partial h} - \left[\frac{\partial \mathbf{K}}{\partial h} \theta + \frac{\partial \mathbf{C}}{\partial h} \dot{\theta} \right], \quad (4.9)$$

gdzie θ oraz $\dot{\theta}$ to odpowiednio temperatura i prędkość zmian temperatury w założonej chwili czasowej. Do rozwiązania tego równania zastosujemy ten sam schemat przyrostowy, który stosowany był do rozwiązywania równania przepływu ciepła (3.11),

$$\frac{d^{t+\Delta t} \theta}{dh} = (1 - \alpha) \frac{d^t \theta}{dh} + \alpha \frac{d^{t+\Delta t} \theta}{dh}, \quad \alpha \in [0, 1], \quad (4.10)$$

$$(4.11)$$

$$\frac{d^{t+\Delta t} \dot{\theta}}{dh} = \frac{1}{\Delta t} \left(\frac{d^{t+\Delta t} \theta}{dh} - \frac{d^t \theta}{dh} \right), \quad (4.12)$$

a po podstawieniu do równania wrażliwości przepływu ciepła (4.8), otrzymujemy

$$\left[\alpha \mathbf{K} + \frac{1}{\Delta t} \mathbf{C} \right] \frac{d^{t+\Delta t} \boldsymbol{\theta}}{dh} = \frac{\partial^{t+\Delta t} \mathbf{P}}{\partial h} - \left[\frac{\partial \mathbf{K}}{\partial h} \boldsymbol{\theta} + \frac{\partial \mathbf{C}}{\partial h} \dot{\boldsymbol{\theta}} \right] + \left[\frac{1}{\Delta t} \mathbf{C} - (1 - \alpha) \mathbf{K} \right] \frac{\partial^t \boldsymbol{\theta}}{\partial h}. \quad (4.13)$$

Dla uproszczenia równanie wrażliwości dla problemu przepływu ciepła można zapisać w postaci.

$${}^{t+\alpha\Delta t} \widehat{\mathbf{K}} \frac{d^{t+\Delta t} \boldsymbol{\theta}}{dh} = \frac{d^{t+\alpha\Delta t} \widehat{\mathbf{P}}}{dh} - \frac{d^{t+\alpha\Delta t} \widehat{\mathbf{K}}}{dh} {}^{t+\Delta t} \boldsymbol{\theta}. \quad (4.14)$$

4.3 Analiza wrażliwości w lepkosprężystości

Stosując metodę DDM różniczkujemy wszystkie elementy składowe układu równań problemu lepkosprężystego (3.42)- (3.53) otrzymując

$$\begin{aligned} \frac{d \mathbf{K}^0}{dh} &= \int_{\Omega} \mathbf{B}^T \frac{d \mathbf{C}^0}{dh} \mathbf{B} d\Omega, \\ \frac{d \tilde{\mathbf{K}}}{dh} &= \left[\int_{\Omega} \mathbf{B}^T \left[\frac{d \tilde{D}}{d\theta} \frac{d\theta}{dh}, \tilde{\mathbf{C}} + \tilde{D} \frac{d \tilde{\mathbf{C}}}{dh} \right] \mathbf{B} d\Omega \right], \\ \frac{d \mathbf{K}^*}{dh} &= \int_{\Omega} \mathbf{B}^T \frac{d \mathbf{C}^*}{d\theta} \frac{d\theta}{dh} \mathbf{B} d\Omega, \\ \frac{d \mathbf{K}_S}{dh} &= \int_{\Omega} \bar{\mathbf{N}}^T \boldsymbol{\alpha}^T \left[\frac{d \mathbf{C}^*}{d\theta} \frac{d\theta}{dh} \boldsymbol{\varepsilon}^{(\theta)} + \mathbf{C}^* \frac{d \boldsymbol{\varepsilon}^{(\theta)}}{d\theta} \frac{d\theta}{dh} \right] \mathbf{B} d\Omega, \\ \frac{d {}^{t+\Delta t} \mathbf{F}^{(\theta)}}{dh} &= \int_{\Omega} \boldsymbol{\alpha}^T \theta_0^{t+\Delta t} \left[\frac{d \mathbf{C}^*}{d\theta} \frac{d\theta}{dh} \boldsymbol{\varepsilon}^{(\theta)} + \mathbf{C}^* \frac{d \boldsymbol{\varepsilon}^{(\theta)}}{d\theta} \frac{d\theta}{dh} \right] \mathbf{B} d\Omega, \\ \frac{d {}^{t+\Delta t} \mathbf{F}^0}{dh} &= \left[\int_{\Omega} \mathbf{B}^T \frac{d \mathbf{C}^0}{dh} \mathbf{B} d\Omega \right] {}^i \mathbf{q} + \mathbf{K}^0 \frac{d {}^t \mathbf{q}}{dh}, \\ \frac{d {}^t \tilde{\mathbf{F}}}{dh} &= \left[\int_{\Omega} \mathbf{B}^T \frac{d}{d\theta} (\tilde{D} \tilde{\mathbf{C}}) \frac{d\theta}{dh} \mathbf{B} d\Omega \right] {}^t \mathbf{q} + \\ &+ \left[\int_{\Omega} \mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B} d\Omega \right] \frac{d {}^t \mathbf{q}}{dh} = \frac{d}{dh} (\tilde{\mathbf{K}} {}^t \mathbf{q}) = \frac{\partial \tilde{\mathbf{K}}}{\partial h} {}^t \mathbf{q} + \tilde{\mathbf{K}} \frac{d {}^t \mathbf{q}}{dh}, \end{aligned}$$

$$\begin{aligned}
\frac{d \Delta \tilde{\mathbf{F}}}{dh} &= \left[\int_{\Omega} \mathbf{B}^T \frac{d}{d\theta} (\tilde{D}\tilde{\mathbf{C}}) \frac{d\theta}{dh} \mathbf{B} d\Omega \right] \Delta \mathbf{q} + \\
&+ \left[\int_{\Omega} \mathbf{B}^T \tilde{D}\tilde{\mathbf{C}} \mathbf{B} d\Omega \right] \frac{d \Delta \mathbf{q}}{dh} = \frac{d}{dh} (\tilde{\mathbf{K}} \Delta \mathbf{q}) = \frac{\partial \tilde{\mathbf{K}}}{\partial h} \Delta \mathbf{q} + \tilde{\mathbf{K}} \frac{d \Delta \mathbf{q}}{dh}, \\
\frac{d {}^t \bar{\mathbf{F}}^{(c)}}{dh} &= \left[\int_{\Omega} \frac{d}{d\theta} (D^{(c)}) \frac{d\theta}{dh} \mathbf{B}^T {}^t \boldsymbol{\sigma}^{(c)} d\Omega \right] + \\
&+ \left[\int_{\Omega} D^{(c)} \mathbf{B}^T \frac{d {}^t \boldsymbol{\sigma}^{(c)}}{d\theta} \frac{d\theta}{dh} d\Omega \right], \\
\frac{d {}^t \mathbf{F}^{(c)}}{dh} &= \frac{d {}^t \bar{\mathbf{F}}^{(c)}}{dh} + \frac{d {}^t \tilde{\mathbf{F}}}{dh}, \\
\frac{d {}^{t+\Delta t} \mathbf{Q}}{dh} &= 0, \\
\frac{d {}^{t+\Delta t} \hat{\mathbf{Q}}}{dh} &= -\frac{d {}^t \mathbf{F}^0}{dh} + \frac{d {}^{t+\Delta t} \mathbf{F}^{(c)}}{dh} + \frac{d {}^{t+\Delta t} \mathbf{F}^{\theta}}{dh}.
\end{aligned} \tag{4.15}$$

Następnie różniczkując układ równań(4.17) po h otrzymujemy

$$\begin{bmatrix} \hat{\mathbf{K}} & 0 \\ {}^{t+\Delta t} \mathbf{K}_S & \mathbf{K}^* \end{bmatrix} \begin{bmatrix} \frac{d {}^{t+\Delta t} \boldsymbol{\theta}}{dh} \\ \frac{d \Delta \mathbf{q}}{dh} \end{bmatrix} = \begin{bmatrix} \frac{d {}^{t+\Delta t} \hat{\mathbf{P}}}{dh} \\ \frac{d {}^{t+\Delta t} \hat{\mathbf{Q}}}{dh} \end{bmatrix} - \begin{bmatrix} \frac{d {}^{t+\Delta t} \hat{\mathbf{K}}}{dh} & 0 \\ \frac{d {}^{t+\Delta t} \mathbf{K}_S}{dh} & \frac{d {}^{t+\Delta t} \mathbf{K}^*}{dh} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \boldsymbol{\theta} \\ \Delta \mathbf{q} \end{bmatrix}. \tag{4.16}$$

Drugi układ równań z (4.16) można zapisać następująco

$$\mathbf{K}^* \frac{d \Delta \mathbf{q}}{dh} = \frac{d {}^{t+\Delta t} \hat{\mathbf{Q}}}{dh} - {}^{t+\Delta t} \mathbf{K}_S \frac{d {}^{t+\Delta t} \boldsymbol{\theta}}{dh} - \frac{d {}^{t+\Delta t} \mathbf{K}_S}{dh} {}^{t+\Delta t} \boldsymbol{\theta}, \tag{4.17}$$

Żeby rozwiązać ten układ równań trzeba najpierw rozwiązać układ (4.17) by otrzymać ${}^{t+\Delta t} \boldsymbol{\theta}$. Następnie rozwiązujemy układ równań problemu wrażliwości (4.16). Pierwszy podukład rozwiązujemy ze względu na $\frac{d {}^{t+\Delta t} \boldsymbol{\theta}}{dh}$, a następnie z drugiego podukładu wyznaczamy $\frac{d \Delta \mathbf{q}}{dh}$, wykorzystując zdekomponowaną macierz \mathbf{K}^* , którą mamy z rozwiązania problemu równowagi (4.17). Prawą stroną układu równań Eq. (4.17) można łatwo wyznaczyć, gdyż:

- macierz sztywności \mathbf{K}^* jest jawną funkcją parametru projektowego h ,
- $\frac{d {}^t \boldsymbol{\theta}}{dh}$, $\frac{d {}^t \mathbf{q}}{dh}$ oraz $\frac{d {}^t \boldsymbol{\sigma}^{(c)}}{dh}$ są znane i tylko te wielkości muszą być przechowywane w pamięci komputera jako dane z poprzedniego kroku.

Akumulacja wrażliwości odbywa się w/g wzoru

$$\frac{d {}^{t+\Delta t}\mathbf{q}}{dh} = \frac{d {}^t\mathbf{q}}{dh} + \frac{d \Delta\mathbf{q}}{dh}. \quad (4.18)$$

Mając wartości wrażliwości na początku kroku czasowego i dodając do niego przyrost wartości wrażliwości wyliczony z równania Eq. (4.16), mamy wartości wrażliwości na końcu kroku. Dla wyliczenia wartości na końcu kroku $\frac{d\boldsymbol{\sigma}^{(c)}}{dh}$ różniczkujemy równanie Eq. (3.34) otrzymując

$$\begin{aligned} \frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh} &= \frac{d D^{(c)}}{d\theta} \frac{d\theta}{dh} {}^t\boldsymbol{\sigma}^{(c)} + D^{(c)} \frac{d {}^t\boldsymbol{\sigma}^{(c)}}{d\theta} \frac{d\theta}{dh} + \\ &+ \frac{d \tilde{D}}{d\theta} \frac{d\theta}{dh} \tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} + \tilde{D} \frac{d \tilde{\mathbf{C}}}{dh} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} + \tilde{D} \tilde{\mathbf{C}} \frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh}. \end{aligned} \quad (4.19)$$

Wszystkie wielkości z prawej strony są znane wartości na początku kroku są przechowywane w pamięci komputera podczas gdy wielkości ${}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}$ oraz $\frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh}$ można łatwo obliczyć ze związków

$${}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} = \mathbf{B} {}^{t+\Delta t}\mathbf{q} - {}^{t+\Delta t}\boldsymbol{\varepsilon}^{(\theta)} \quad (4.20)$$

$$\frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh} = \mathbf{B} \frac{d {}^{t+\Delta t}\mathbf{q}}{dh} - \frac{d {}^{t+\Delta t}\boldsymbol{\varepsilon}^{(\theta)}}{dh} = \mathbf{B} \frac{d {}^{t+\Delta t}\mathbf{q}}{dh} - \frac{d}{dh}(\boldsymbol{\alpha}[\tilde{\mathbf{N}}\boldsymbol{\theta} - \theta_0]). \quad (4.21)$$

Tak więc mając (4.19), można łatwo otrzymać wartości $\frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh}$. Warto zauważyć, że konieczność posiadania wartości $\frac{d\mathbf{q}}{dh}$ praktycznie uniemożliwia zastosowanie metody systemu sprzężonego do obliczania wrażliwości.

W przypadku korzystania z iteracyjnej wersji układu równań w problemie równowagi (3.61) analiza wrażliwości również wymagać będzie iteracyjnej wersji równań. Zróżniczkowanie równania (3.61) prowadzi do otrzymania poniższych związków

$$\begin{aligned} \mathbf{K}^0 \frac{d \Delta\mathbf{q}}{dh} &= -\frac{d \mathbf{K}^0}{dh} \Delta\mathbf{q} + \frac{d {}^{t+\Delta t}\hat{\mathbf{Q}}}{dh} - {}^{t+\Delta t}\mathbf{K}_S \frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh} + \\ &- \frac{d {}^{t+\Delta t}\mathbf{K}_S}{dh} {}^{t+\Delta t}\boldsymbol{\theta} - \frac{d \Delta\tilde{\mathbf{F}}}{dh}, \end{aligned} \quad (4.22)$$

Mając wszystkie niezbędne równania można teraz przedstawić algorytm obliczania wrażliwości na kroku czasowym $[t, t + \Delta t]$ (Box 2).

- (a) Rozwiązanie problemu niestacjonarnego przepływu ciepła dla uzyskania pola temperatur oraz wrażliwości temperatur dla czasu $t + \Delta t$. Potrzebne do dalszych obliczeń są również wartości $A^{(t+\Delta t\theta)}$, wraz z wrażliwościami.
- (b) na podstawie znanych wartości $({}^t\mathbf{q}, {}^t\boldsymbol{\sigma}^{(c)})$ dla czasu t z układu równań (4.17) wyznaczenie $\Delta\mathbf{q}$.
- (c) Znając ${}^{t+\Delta t}\boldsymbol{\theta}$, $\frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh}$, $\Delta\mathbf{q}$ oraz $\frac{d {}^t\mathbf{q}}{dh}$, $\frac{d {}^t\boldsymbol{\sigma}^{(c)}}{dh}$, obliczenie wartości wrażliwości przemieszczeń (4.16) wykorzystując zdekomponowaną w pkt. (b) macierz \mathbf{K}^* .
- (d) Uaktualnienie wartości

$$\begin{aligned}
{}^{t+\Delta t}\boldsymbol{\theta} &= {}^t\boldsymbol{\theta} + \Delta\boldsymbol{\theta}, \\
{}^{t+\Delta t}\mathbf{q} &= {}^t\mathbf{q} + \Delta\mathbf{q}, \\
{}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} &= \mathbf{B} {}^{t+\Delta t}\mathbf{q} - \boldsymbol{\varepsilon}^{(\theta)}, \\
{}^{t+\Delta t}\boldsymbol{\sigma}^{(c)} &= D^{(c)t} \boldsymbol{\sigma}^{(c)} + \tilde{D} \tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}, \\
{}^{t+\Delta t}\boldsymbol{\sigma} &= \mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}
\end{aligned} \tag{4.23}$$

oraz wrażliwości

$$\begin{aligned}
\frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh} &= \frac{d {}^t\boldsymbol{\theta}}{dh} + \frac{d \Delta\boldsymbol{\theta}}{dh}, \\
\frac{d {}^{t+\Delta t}\mathbf{q}}{dh} &= \frac{d {}^t\mathbf{q}}{dh} + \frac{d \Delta\mathbf{q}}{dh}, \\
\frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh} &= \mathbf{B} \frac{d {}^{t+\Delta t}\mathbf{q}}{dh} - \frac{d {}^{t+\Delta t}\boldsymbol{\varepsilon}^{(\theta)}}{dh}, \\
\frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh} &= \text{z równania Eq. (4.19),} \\
\frac{d {}^{t+\Delta t}\boldsymbol{\sigma}}{dh} &= \frac{\partial \mathbf{C}^0}{\partial h} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} + \mathbf{C}^0 \frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh} - \frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh} \frac{d\theta}{dh}
\end{aligned} \tag{4.24}$$

- (e) Przejście do następnego kroku ${}^{t+\Delta t}\boldsymbol{\theta}$, $\frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh}$, ${}^{t+\Delta t}\mathbf{q}$, ${}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}$ and $\frac{d {}^{t+\Delta t}\mathbf{q}}{dh}$, $\frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh}$ zachowane w pamięci.

Box 2. Nieiteracyjny algorytm obliczania wrażliwości dla typowego kroku czasowego $[t, t + \Delta t]$

4.4 Wrażliwość na parametry kształtu

W problemie wrażliwości na parametry kształtu istnieją zmienne projektowe opisujące kształt konstrukcji. Współrzędne węzłów siatki MES zależą od tych parametrów. Do rozwiązania tego problemu zastosowana będzie metoda zwana w języku angielskim *Domain Parameterization Approach (DPA)*. Koncepcja metody DPA jest podobna do podejścia izoparametrycznego, co uwidacznia Rys. 4.1. Wyróżniamy tu niezależną od parametrów projektowych konfigurację odniesienia C^r oraz zależną od nich konfigurację aktualną C^t . Dla uproszczenia przyjmijmy jeden parametr kształtu h (bez większych

kłopotów można by przyjąć ich więcej). Mamy zatem

$$\mathbf{x} = f(\boldsymbol{\xi}, h), \quad \mathbf{x} \in \Omega, \quad \boldsymbol{\xi} \in \Omega^r, \quad (4.25)$$

$$\mathbf{F} = \frac{d\mathbf{x}}{d\boldsymbol{\xi}}, \quad J = \det \mathbf{F}, \quad d\Omega = J d\Omega^r. \quad (4.26)$$

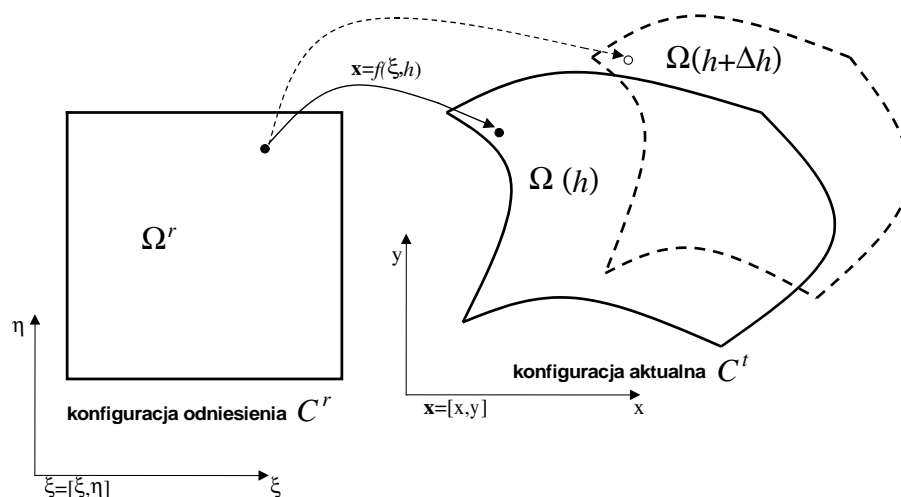
Wszystkie całki liczone podczas analizy muszą być obliczane po obszarze konfiguracji aktualnej. Mając związek (4.26) można jednak przetransformować te całki do obszaru konfiguracji odniesienia. (4.27). Otrzymamy więc

$$\int_{\Omega} \dots d\Omega = \int_{\Omega^r} \dots J d\Omega^r \quad (4.27)$$

Wrażliwość kształtu całki funkcji f obliczana zatem będzie następująco

$$\frac{d}{dh} \int_{\Omega} f d\Omega = \int_{\Omega^r} \left[\frac{df}{dh} J + f \frac{dJ}{dh} \right] d\Omega^r \quad (4.28)$$

Układ równań problemu wrażliwości kształtu jest taki sam, jak (4.16), lecz z użyciem związku (4.28) przy obliczaniu całek. Zatem definicje poszczególnych elementów układu



Rysunek 4.1: Konfiguracja aktualna $\Omega(h)$ i konfiguracja odniesienia Ω^r

równań będą wyglądały następująco

$$\begin{aligned}
\mathbf{K}^0 &= \int_{\Omega} \frac{d\mathbf{B}^T}{dh} \mathbf{C}^0 \mathbf{B} + \mathbf{B}^T \mathbf{C}^0 \frac{d\mathbf{B}}{dh} + \mathbf{B}^T \mathbf{C}^0 \mathbf{B} \frac{dJ}{dh} d\Omega, & (4.29) \\
\frac{d\mathbf{K}^0}{dh} &= \int_{\Omega^r} [\mathbf{B}^T \mathbf{C}^0 \mathbf{B}] \frac{dJ}{dh} + \frac{d}{dh} [\mathbf{B}^T \mathbf{C}^0 \mathbf{B}] J d\Omega^r, \\
\frac{d\tilde{\mathbf{K}}}{dh} &= \int_{\Omega^r} [\mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B}] \frac{dJ}{dh} + \frac{d}{dh} [\mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B}] J d\Omega^r, \\
\frac{d\mathbf{K}^*}{dh} &= \int_{\Omega^r} [\mathbf{B}^T \mathbf{C}^* \mathbf{B}] \frac{dJ}{dh} + \frac{d}{dh} [\mathbf{B}^T \mathbf{C}^* \mathbf{B}] J d\Omega^r, \\
\frac{d\mathbf{K}_S}{dh} &= \int_{\Omega^r} \bar{\mathbf{N}}^T \boldsymbol{\alpha}^T \frac{d}{dh} [\mathbf{C}^* \boldsymbol{\varepsilon}^{(\theta)} \mathbf{B} J] d\Omega^r, \\
\frac{d\mathbf{F}^{(\theta)}}{dh} &= \int_{\Omega^r} \boldsymbol{\alpha}^T \theta_0 \frac{d}{dh} [{}^{t+\Delta t} \mathbf{C}^* \boldsymbol{\varepsilon}^{(\theta)} \mathbf{B} J] d\Omega^r, \\
\frac{d {}^t \mathbf{F}^0}{dh} &= \left[\int_{\Omega^r} [\mathbf{B}^T \mathbf{C}^0 \mathbf{B}] \frac{dJ}{dh} + \frac{d}{dh} [\mathbf{B}^T \mathbf{C}^0 \mathbf{B}] J d\Omega^r \right] {}^i \mathbf{q} + \mathbf{K}^0 \frac{d {}^t \mathbf{q}}{dh}, \\
\frac{d {}^t \tilde{\mathbf{F}}}{dh} &= \left[\int_{\Omega^r} \frac{d}{dh} (\mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B} J) d\Omega^r \right] {}^t \mathbf{q} \\
&+ \left[\int_{\Omega^r} \mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B} J d\Omega^r \right] \frac{d {}^t \mathbf{q}}{dh}, \\
& & (4.30) \\
\frac{d \Delta \tilde{\mathbf{F}}}{dh} &= \left[\int_{\Omega^r} \frac{d}{dh} (\mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B} J) d\Omega^r \right] \Delta \mathbf{q} \\
&+ \left[\int_{\Omega^r} \mathbf{B}^T \tilde{D} \tilde{\mathbf{C}} \mathbf{B} J d\Omega^r \right] \frac{d \Delta \mathbf{q}}{dh}, \\
\frac{d {}^t \bar{\mathbf{F}}^{(c)}}{dh} &= \int_{\Omega^r} \frac{d}{dh} (D^{(c)}, \mathbf{B}^T {}^t \boldsymbol{\sigma}^{(c)} J) d\Omega^r, \\
\frac{d {}^t \mathbf{F}^{(c)}}{dh} &= \frac{d {}^t \bar{\mathbf{F}}^{(c)}}{dh} + \frac{d {}^t \tilde{\mathbf{F}}}{dh}, \\
\frac{d {}^{t+\Delta t} \mathbf{Q}}{dh} &= 0, \\
\frac{d {}^{t+\Delta t} \hat{\mathbf{Q}}}{dh} &= -\frac{d {}^t \mathbf{F}^0}{dh} + \frac{d {}^{t+\Delta t} \mathbf{F}^{(c)}}{dh} + \frac{d {}^{t+\Delta t} \mathbf{F}^\theta}{dh}.
\end{aligned}$$

4.5 Wrażliwość parametryczna w modelu zniszczenia

Do wyznaczenia równań wrażliwości zastosujemy metodę DDM. Różniczkując równanie (3.97) mamy

$$\mathbf{K}_T \frac{d\Delta \mathbf{q}}{dh} = \frac{\partial \mathbf{P}}{\partial h} - \frac{d\mathbf{F}}{dh} - \frac{d\mathbf{K}_T}{dh} \Delta \mathbf{q}, \quad (4.31)$$

Powyższe równanie pozwala na wyznaczenie wrażliwości przyrostu przemieszczeń. Addytywność pochodnych sprawia, że przyrost ten może być akumulowany, co pozwoli otrzymać wrażliwość przemieszczeń. Poszczególne wyrazy równania (4.31) nie posiadają indeksu iteracji, gdyż są to wielkości wyznaczone na końcu procesu iteracyjnego równania równowagi, po uzyskaniu żądanej zbieżności (3.97). Poniżej zamieścimy definicje poszczególnych wyrazów układu równań.

$$\mathbf{K}_T = \int_{\Omega} \mathbf{B}^T \mathbf{C}_D^* \mathbf{B} d\Omega, \quad (4.32)$$

$$\frac{d\mathbf{K}_T}{dh} = \int_{\Omega} \mathbf{B}^T \frac{\partial \mathbf{C}_D^*}{\partial h} \mathbf{B} d\Omega, \quad (4.33)$$

$$\frac{d\mathbf{F}}{dh} = \int_{\Omega} \mathbf{B}^T \frac{\partial \mathbf{C}_D}{\partial h} \mathbf{B}^i \mathbf{q} + \mathbf{B}^T \mathbf{C}_D \mathbf{B} \frac{d\mathbf{q}}{dh} d\Omega. \quad (4.34)$$

4.5.1 Wrażliwość parametryczna w modelu termolepkosprężystym ze zniszczeniem

Stosując jak poprzednio DDM do wyznaczenia równań wrażliwości, różniczkujemy układ równań (3.132), otrzymujemy

$$\begin{bmatrix} {}^{t+\alpha\Delta t} \widehat{\mathbf{K}} & 0 \\ {}^{t+\Delta t} \mathbf{K}_S & \mathbf{K}_T \end{bmatrix} \begin{bmatrix} \frac{d^{t+\Delta t} \boldsymbol{\theta}}{dh} \\ \frac{d\Delta \mathbf{q}}{dh} \end{bmatrix} = \begin{bmatrix} \frac{d^{t+\alpha\Delta t} \widehat{\mathbf{P}}}{dh} \\ \frac{d^{t+\Delta t} \widehat{\mathbf{Q}}}{dh} \end{bmatrix} - \begin{bmatrix} \frac{d^{t+\alpha\Delta t} \widehat{\mathbf{K}}}{dh} & 0 \\ \frac{d^{t+\Delta t} \mathbf{K}_S}{dh} & \frac{d^{t+\Delta t} \mathbf{K}_T}{dh} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t} \boldsymbol{\theta} \\ \Delta \mathbf{q} \end{bmatrix}. \quad (4.35)$$

Poszczególne składowe powyższego układu równań dla termolepkosprężystego modelu ze zniszczeniem obliczone poprzez zróżniczkowanie składowych (4.37)-(4.44) zdefiniowane są następująco

$$\frac{d^{t+\Delta t} \widetilde{\boldsymbol{\varepsilon}}}{dh} = \mathbf{B} \frac{d\mathbf{q}}{dh} - \frac{d\boldsymbol{\alpha}}{dh} \mathbf{N} \boldsymbol{\theta} - \boldsymbol{\alpha} \mathbf{N} \frac{d\boldsymbol{\theta}}{dh} + \frac{d\boldsymbol{\alpha} \boldsymbol{\theta}_0}{dh}, \quad (4.36)$$

$$\begin{aligned} \frac{d\mathbf{K}_T}{dh} &= \int_{\Omega} \mathbf{B}^T \left[(1 - {}^{t+\Delta t} d) \frac{d\overline{\mathbf{C}}^*}{dh} - \frac{d^{t+\Delta t} d}{dh} \overline{\mathbf{C}}^* \right] \mathbf{B} d\Omega - \\ &+ \int_{\Omega} \mathbf{B}^T \left[\frac{\partial^2 {}^{t+\Delta t} d}{\partial \boldsymbol{\varepsilon} \partial h} \left(\overline{\mathbf{C}}^* {}^{t+\Delta t} \widetilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t} D^{(c)} {}^t \boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} d\Omega - \end{aligned} \quad (4.37)$$

$$+ \int_{\Omega} \mathbf{B}^T \left[\frac{\partial^{t+\Delta t} d}{\partial \boldsymbol{\varepsilon}} \frac{d}{dh} \left(\overline{\mathbf{C}}^{*t+\Delta t} \tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t} D^{(c)} {}^t \boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} d\Omega.$$

$$\begin{aligned} \frac{d}{dh} \left(\overline{\mathbf{C}}^{*t+\Delta t} \tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t} D^{(c)} {}^t \boldsymbol{\sigma}^{(c)} \right) &= \frac{d\overline{\mathbf{C}}^*}{dh} {}^{t+\Delta t} \tilde{\boldsymbol{\varepsilon}} + \overline{\mathbf{C}}^* \frac{d^{t+\Delta t} \tilde{\boldsymbol{\varepsilon}}}{dh} - \frac{d^{t+\Delta t} D^{(c)}}{d\theta} \frac{d\theta}{dh} {}^t \boldsymbol{\sigma}^{(c)} - \\ &+ {}^{t+\Delta t} D^{(c)} \frac{d^t \boldsymbol{\sigma}^{(c)}}{dh}, \end{aligned} \quad (4.38)$$

$$\frac{\partial \mathbf{K}_s}{\partial h} = \int_{\Omega} \mathbf{B}^T {}^{t+\Delta t} d^i \frac{\partial \boldsymbol{\alpha}}{\partial h} \mathbf{N} - \mathbf{B}^T \frac{d^{t+\Delta t} d^i}{dh} \boldsymbol{\alpha} \mathbf{N} d\Omega, \quad (4.39)$$

$$\begin{aligned} \frac{d\mathbf{F}}{dh} &= \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t} d) \frac{\partial \overline{\mathbf{C}}^*}{\partial h} \mathbf{B} {}^{t+\Delta t} \mathbf{q} - \mathbf{B}^T \frac{d^{t+\Delta t} d^i}{dh} \overline{\mathbf{C}}^* \mathbf{B} {}^{t+\Delta t} \mathbf{q} d\Omega + \\ &+ \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t} d^i) \overline{\mathbf{C}}^* \mathbf{B} \frac{d^{t+\Delta t} \mathbf{q}}{dh} d\Omega, \end{aligned} \quad (4.40)$$

$$\begin{aligned} \frac{d\mathbf{F}_c}{dh} &= \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t} d) \frac{d^{t+\Delta t} D^{(c)}}{d\theta} \frac{d\theta}{dh} \boldsymbol{\sigma}^{(c)} - \mathbf{B}^T \frac{\partial^{t+\Delta t} d}{\partial h} {}^{t+\Delta t} D^{(c)} {}^t \boldsymbol{\sigma}^{(c)} d\Omega + \\ &+ \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t} d) {}^{t+\Delta t} D^{(c)} \frac{d\boldsymbol{\sigma}^{(c)}}{dh} d\Omega, \end{aligned} \quad (4.41)$$

$$\mathbf{F}_{\theta} = \int_{\Omega} \mathbf{B}^T (1 - {}^{t+\Delta t} d) \overline{\mathbf{C}}^* \frac{\partial \boldsymbol{\alpha}}{\partial h} \theta_0 + \mathbf{B}^T (1 - {}^{t+\Delta t} d) \frac{\partial \overline{\mathbf{C}}^*}{\partial h} \boldsymbol{\alpha} \theta_0 d\Omega - \quad (4.42)$$

$$+ \int_{\Omega} \mathbf{B}^T \frac{d^{t+\Delta t} d}{dh} \overline{\mathbf{C}}^* \boldsymbol{\alpha} \theta_0 d\Omega, \quad (4.43)$$

$$\frac{d\widehat{\mathbf{Q}}}{dh} = \frac{d^{t+\Delta t} \mathbf{Q}}{dh} - \frac{d\mathbf{F}}{dh} - \frac{d\mathbf{F}_c}{dh} - \frac{d\mathbf{F}_{\theta}}{dh}. \quad (4.44)$$

4.5.2 Wrażliwość na parametry kształtu w modelu ze zniszczeniem

Wrażliwość na parametry kształtu dla problemu termolepkosprężystego ze zniszczeniem znajdujemy rozwiązując ten sam układ równań (4.35), co dla wrażliwości parametrycznej. Inne są jednak definicje składowych układu równań ze względu na to, że wielkości zależne od parametrów projektowych są inne niż w problemie wrażliwości parametrycznej. Przykładowo macierze sztywności materiałowej nie zależą od zmiennych projektowych parametryzujących kształt konstrukcji.

$${}^{t+\Delta t} \tilde{\boldsymbol{\varepsilon}}^{(i)} = \frac{d\mathbf{B}}{dh} \mathbf{q} + \mathbf{B} \frac{d\mathbf{q}}{dh} - \boldsymbol{\alpha} \mathbf{N} \frac{d\boldsymbol{\theta}}{dh} + \boldsymbol{\alpha} \theta_0, \quad (4.45)$$

$$\begin{aligned} \frac{d\mathbf{K}_s}{dh} &= - \int_{\Omega^r} \frac{d\mathbf{B}^T}{dh} (1 - {}^{t+\Delta t} d) \boldsymbol{\alpha} \mathbf{N} J d\Omega^r + \int_{\Omega^r} \mathbf{B}^T \frac{d^{t+\Delta t} d}{dh} \boldsymbol{\alpha} \mathbf{N} J d\Omega^r + \\ &+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t} d) \boldsymbol{\alpha} \mathbf{N} \frac{d}{dh} J d\Omega^r, \end{aligned} \quad (4.46)$$

$$\frac{d\mathbf{K}_T}{dh} = \int_{\Omega^r} \mathbf{B}^T \left[(1 - {}^{t+\Delta t}d) \frac{d\bar{\mathbf{C}}^*}{dh} - \frac{d^{t+\Delta t}d}{dh} \bar{\mathbf{C}}^* \right] \mathbf{B} J d\Omega^r - \quad (4.47)$$

$$+ \int_{\Omega^r} \mathbf{B}^T \left[\frac{\partial^2 {}^{t+\Delta t}d}{\partial \boldsymbol{\varepsilon} \partial h} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} J d\Omega^r -$$

$$+ \int_{\Omega^r} \mathbf{B}^T \left[\frac{\partial {}^{t+\Delta t}d}{\partial \boldsymbol{\varepsilon}} \frac{d}{dh} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} J d\Omega^r +$$

$$+ \int_{\Omega^r} \frac{d\mathbf{B}^T}{dh} \left[(1 - {}^{t+\Delta t}d) \bar{\mathbf{C}}^* - \frac{\partial {}^{t+\Delta t}d}{\partial \boldsymbol{\varepsilon}} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} J d\Omega^r$$

$$+ \int_{\Omega^r} \mathbf{B}^T \left[(1 - {}^{t+\Delta t}d) \bar{\mathbf{C}}^* - \frac{\partial {}^{t+\Delta t}d}{\partial \boldsymbol{\varepsilon}} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \frac{d\mathbf{B}}{dh} J d\Omega^r$$

$$+ \int_{\Omega^r} \mathbf{B}^T \left[(1 - {}^{t+\Delta t}d) \bar{\mathbf{C}}^* - \frac{\partial {}^{t+\Delta t}d}{\partial \boldsymbol{\varepsilon}} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) \right] \mathbf{B} \frac{d}{dh} J d\Omega^r$$

$$\frac{d}{dh} \left(\bar{\mathbf{C}}^* {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \right) = \frac{d\bar{\mathbf{C}}^*}{dh} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} + \bar{\mathbf{C}}^* \frac{d^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh} - \frac{d^{t+\Delta t}D^{(c)}}{d\theta} \frac{d\theta}{dh} {}^t\boldsymbol{\sigma}^{(c)} - \quad (4.48)$$

$$+ {}^{t+\Delta t}D^{(c)} \frac{d^t\boldsymbol{\sigma}^{(c)}}{dh}.$$

$$\frac{d\mathbf{F}}{dh} = \int_{\Omega^r} \frac{d\mathbf{B}^T}{dh} (1 - {}^{t+\Delta t}d^i) \bar{\mathbf{C}}^* \mathbf{B}^{t+\Delta t} \mathbf{q} J d\Omega^r - \int_{\Omega^r} \mathbf{B}^T \frac{d^{t+\Delta t}d^i}{dh} \bar{\mathbf{C}}^* \mathbf{B}^{t+\Delta t} \mathbf{q} J d\Omega^r + \quad (4.49)$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \frac{d\bar{\mathbf{C}}^*}{dh} \mathbf{B}^{t+\Delta t} \mathbf{q} J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \bar{\mathbf{C}}^* \frac{d\mathbf{B}}{dh} \mathbf{B}^{t+\Delta t} \mathbf{q} J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \bar{\mathbf{C}}^* \mathbf{B} \frac{d^{t+\Delta t}\mathbf{q}}{dh} J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \bar{\mathbf{C}}^* \mathbf{B}^{t+\Delta t} \mathbf{q} \frac{d}{dh} J d\Omega^r,$$

$$\frac{d\mathbf{F}_\theta}{dh} = \int_{\Omega^r} \frac{d\mathbf{B}^T}{dh} (1 - {}^{t+\Delta t}d^i) \bar{\mathbf{C}}^* \boldsymbol{\alpha} \theta_0 J d\Omega^r - \quad (4.50)$$

$$+ \int_{\Omega^r} \mathbf{B}^T \frac{d^{t+\Delta t}d^i}{dh} \bar{\mathbf{C}}^* \boldsymbol{\alpha} \theta_0 J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \frac{d\bar{\mathbf{C}}^*}{dh} \boldsymbol{\alpha} \theta_0 J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \bar{\mathbf{C}}^* \boldsymbol{\alpha} \theta_0 \frac{d}{dh} J d\Omega^r,$$

$$\frac{d\mathbf{F}_c}{dh} = \int_{\Omega^r} \frac{d\mathbf{B}^T}{dh} (1 - {}^{t+\Delta t}d^i) {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} J d\Omega^r - \quad (4.51)$$

$$- \int_{\Omega^r} \mathbf{B}^T \frac{d^{t+\Delta t}d^i}{dh} {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) \frac{d^{t+\Delta t}D^{(c)}}{d\theta} \frac{d\theta}{dh} {}^t\boldsymbol{\sigma}^{(c)} J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) {}^{t+\Delta t}D^{(c)} \frac{d^t\boldsymbol{\sigma}^{(c)}}{dh} J d\Omega^r +$$

$$+ \int_{\Omega^r} \mathbf{B}^T (1 - {}^{t+\Delta t}d^i) {}^{t+\Delta t}D^{(c)} {}^t\boldsymbol{\sigma}^{(c)} \frac{d}{dh} J d\Omega^r,$$

$$\frac{d^{t+\Delta t}\mathbf{Q}}{dh} = \int_{\Omega^r} (\hat{\mathbf{t}}\mathbf{f} + \Delta\hat{\mathbf{t}})^T \mathbf{N} \frac{d}{dh} J d\Omega^r + \int_{\partial\Omega_\sigma^r} (\hat{\mathbf{t}}\mathbf{t} + \Delta\hat{\mathbf{t}})^T \mathbf{N} \frac{d}{dh} J d\partial\Omega_\sigma^r, \quad (4.52)$$

$$\frac{d\hat{\mathbf{Q}}}{dh} = \frac{d^{t+\Delta t}\mathbf{Q}}{dh} - \frac{d\mathbf{F}}{dh} - \frac{d\mathbf{F}_c}{dh} - \frac{d\mathbf{F}_\theta}{dh}. \quad (4.53)$$

Algorytm wyznaczania wrażliwości parametrycznej oraz kształtu dla przemieszczeń i naprężeń zawiera Box 3.

- (a) Rozwiązanie problemu niestacjonarnego przepływu ciepła dla uzyskanie pola temperatur oraz wrażliwości temperatur dla czasu $t + \Delta t$. Potrzebne do dalszych obliczeń są również wartości $A^{(t+\Delta t)\theta}$ wraz z wrażliwościami.
- (b) na podstawie znanych wartości $({}^t\mathbf{q}, {}^t\boldsymbol{\sigma}^{(c)})$ dla czasu t z układu równań (3.132) wyznaczenie $\Delta\mathbf{q}$.
- (c) Znając ${}^{t+\Delta t}\boldsymbol{\theta}$, $\frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh}$, $\Delta\mathbf{q}$ oraz, $\frac{d {}^t\mathbf{q}}{dh}$, $\frac{d {}^t\boldsymbol{\sigma}^{(c)}}{dh}$, obliczenie wartości wrażliwości przemieszczeń (4.35), wykorzystując zdekomponowaną w pkt. (b) macierz \mathbf{K}_T .

- (d) Uaktualnienie wartości

$$\begin{aligned}
 {}^{t+\Delta t}\boldsymbol{\theta} &= {}^t\boldsymbol{\theta} + \Delta\boldsymbol{\theta}, \\
 {}^{t+\Delta t}\mathbf{q} &= {}^t\mathbf{q} + \Delta\mathbf{q}, \\
 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} &= \mathbf{B} {}^{t+\Delta t}\mathbf{q} - \boldsymbol{\varepsilon}^{(\theta)}, \\
 {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)} &= D^{(c)t}\boldsymbol{\sigma}^{(c)} + \tilde{D}\tilde{\mathbf{C}} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}, \\
 {}^{t+\Delta t}\boldsymbol{\sigma} &= (1 - {}^{t+\Delta t}d)\mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}
 \end{aligned} \tag{4.54}$$

oraz wrażliwości

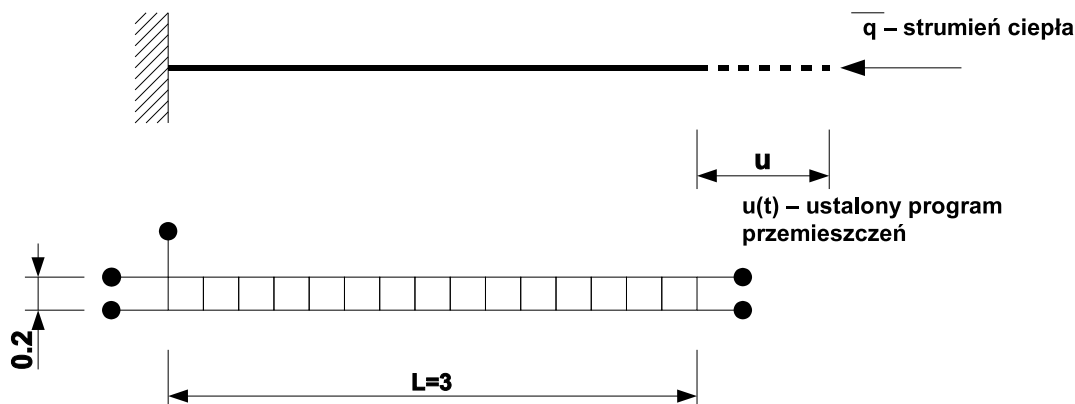
$$\begin{aligned}
 \frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh} &= \frac{d {}^t\boldsymbol{\theta}}{dh} + \frac{d \Delta\boldsymbol{\theta}}{dh}, \\
 \frac{d {}^{t+\Delta t}\mathbf{q}}{dh} &= \frac{d {}^t\mathbf{q}}{dh} + \frac{d \Delta\mathbf{q}}{dh}, \\
 \frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh} &= \mathbf{B} \frac{d {}^{t+\Delta t}\mathbf{q}}{dh} - \frac{d {}^{t+\Delta t}\boldsymbol{\varepsilon}^{(\theta)}}{dh}, \\
 \frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh} &= \text{z równania Eq. (4.19)}, \\
 \frac{d {}^{t+\Delta t}\boldsymbol{\sigma}}{dh} &= (1 - {}^{t+\Delta t}d) \frac{\partial \mathbf{C}^0}{\partial h} {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} + \mathbf{C}^0 \frac{d {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}}}{dh} - \frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{d\theta} \frac{d\theta}{dh} - \\
 &+ \frac{d {}^{t+\Delta t}d}{dh} \mathbf{C}^0 {}^{t+\Delta t}\tilde{\boldsymbol{\varepsilon}} - {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}.
 \end{aligned} \tag{4.56}$$

$$\tag{4.57}$$

- (e) Przejście do następnego kroku ${}^{t+\Delta t}\boldsymbol{\theta}$, $\frac{d {}^{t+\Delta t}\boldsymbol{\theta}}{dh}$, ${}^{t+\Delta t}\mathbf{q}$, ${}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}$ i $\frac{d {}^{t+\Delta t}\mathbf{q}}{dh}$, $\frac{d {}^{t+\Delta t}\boldsymbol{\sigma}^{(c)}}{dh}$ zachowane w pamięci.

Box 3. Nieiteracyjny algorytm obliczania wrażliwości typowego kroku czasowego $[t, t + \Delta t]$ termolepkosprężystego modelu materiału ze zniszczeniem

4.6 Przykład numeryczny - relaksacja naprężeń w pręcie



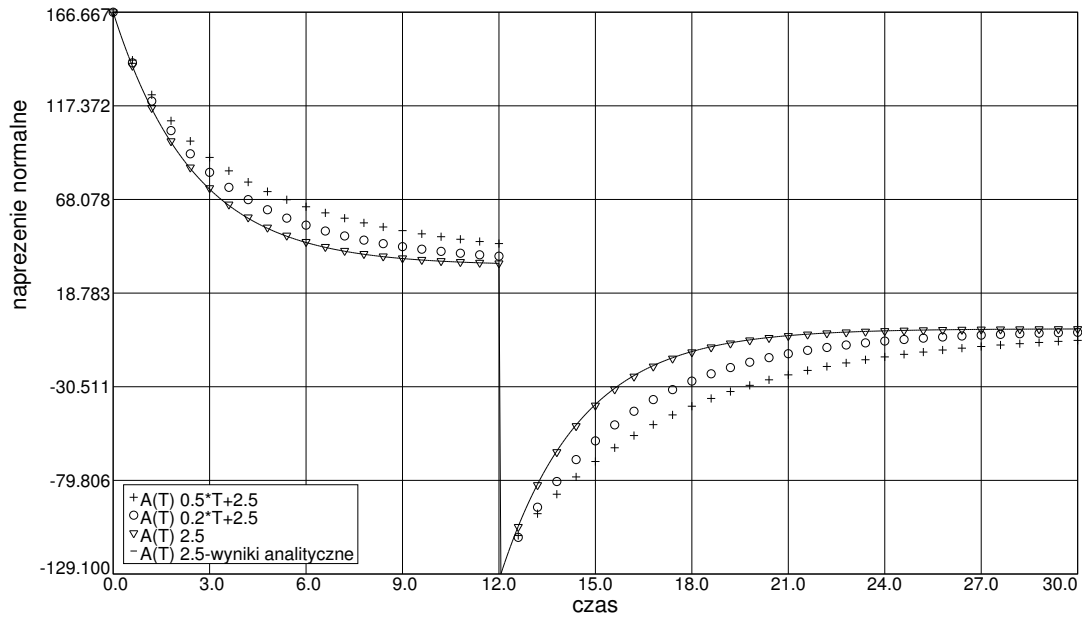
Rysunek 4.2: Schemat pręta. Model pręta składający się z 15 czterowęzłowych elementów typu Lagrange

Model pręta stanowi 15 czterowęzłowych elementów typu Lagrange w płaskim stanie naprężenia. Schemat przedstawia Rys. 4.2. Geometryczne obciążenie końca pręta przemieszczeniem $u_0 = 1$ zostało zdjęte po czasie $\tau_0 = 12s$. Zastosowano w tym teście 200 równych kroków czasowych. Użyto następujących wartości stałych materiałowych $E^0 = 500$, $E^\infty = 100$, $\varrho = 1$, $\nu = 0.3$, $\kappa = \frac{\lambda}{\rho c} = 1$, wartość strumienia ciepła na swobodnym końcu pręta wynosi $\hat{q} = 5$. Funkcja zależności termicznej $A(\theta)$ z równania Eq. 3.18 przyjmuje wartość niezależną od temperatury $A(\theta) = 2.5$ a następnie wartość sprzęgającą problem mechaniczny z przepływem ciepła w postaci $A(\theta) = 0.2 \cdot T + 2.5$ i $A(\theta) = 0.5 \cdot T + 2.5$. Analityczne rozwiązanie tego problemu staje się bardzo trudne nawet dla tak prostego pręta. Dlatego przedstawione tu analityczne rozwiązanie problemu podane zostało dla przypadku pierwszego tj. $A(\theta) = 2.5$ czyli niezależności problemu mechanicznego od rozkładu temperatury (por. [45])

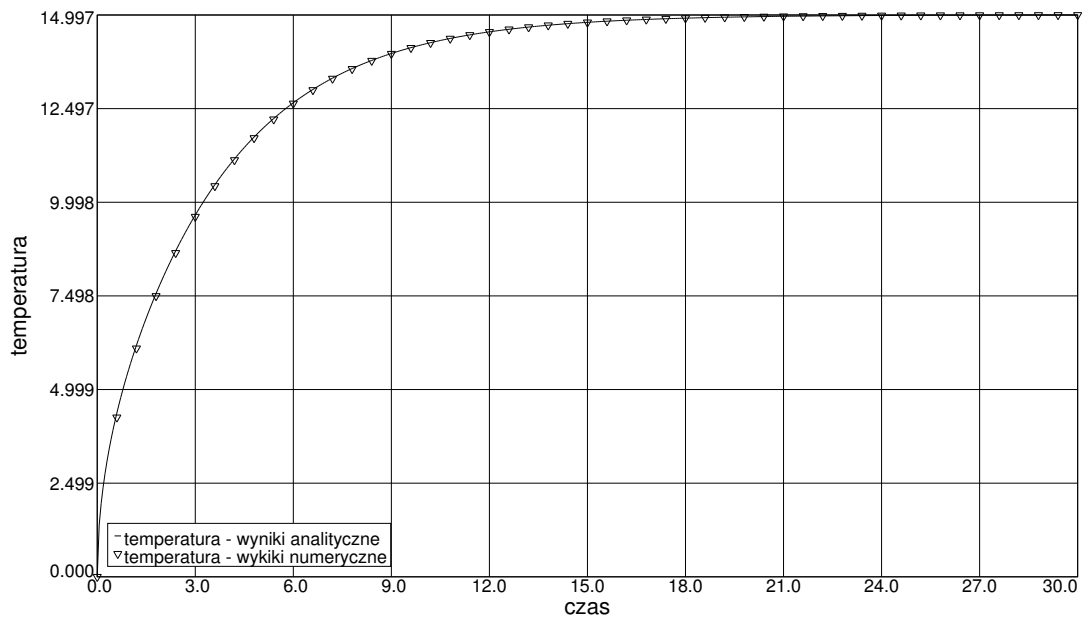
$$u(x, \tau) = \frac{u_0 x}{l} [H(0) - H(\tau_0)]. \quad (4.58)$$

$$\sigma(x, \tau) = \frac{u_0}{l} \cdot \left[E^\infty + \tilde{E} \cdot \exp\left(-\frac{\tau}{\varrho}\right) \right] - H(\tau_0) \frac{u_0}{l} \cdot \left[E^\infty + \tilde{E} \cdot \exp\left(-\frac{\tau - \tau_0}{\varrho}\right) \right] \quad (4.59)$$

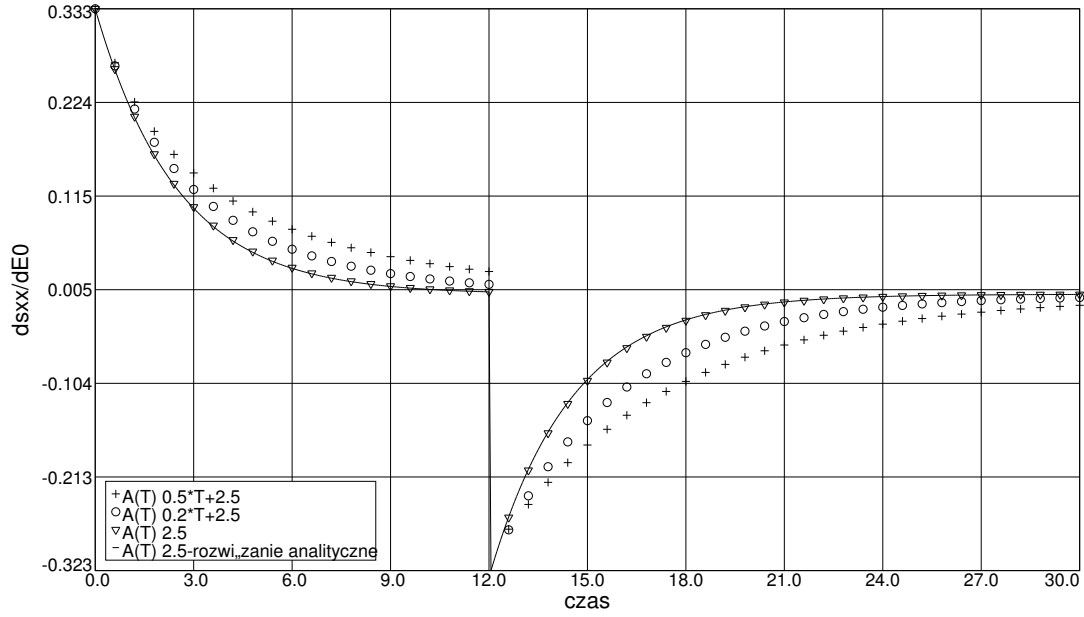
gdzie $H(\tau)$ jest funkcją Heavisite. Na rysunkach 4.3-4.8 obserwujemy bardzo dobrą zbieżność wyników numerycznych z analitycznymi. W przypadku zależnym od temperatury widać zbieżność rezultatów z wartościami dla problemu niezależnego $A(\theta)$, gdzie wpływ temperatury staje się coraz mniejszy. Zmniejszając wartość współczynnika termicznego, obserwujemy szybszą relaksację naprężeń. Rezultaty numeryczne zamieszczone na rys Fig. 4.3 potwierdzają ten fakt.



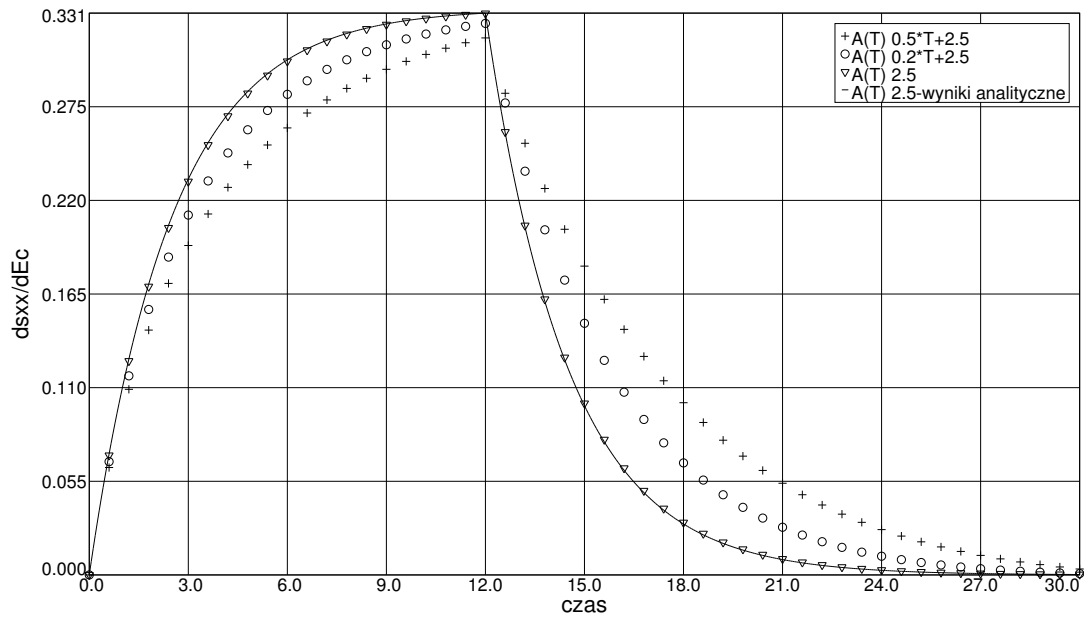
Rysunek 4.3: Relaksacja naprężeń



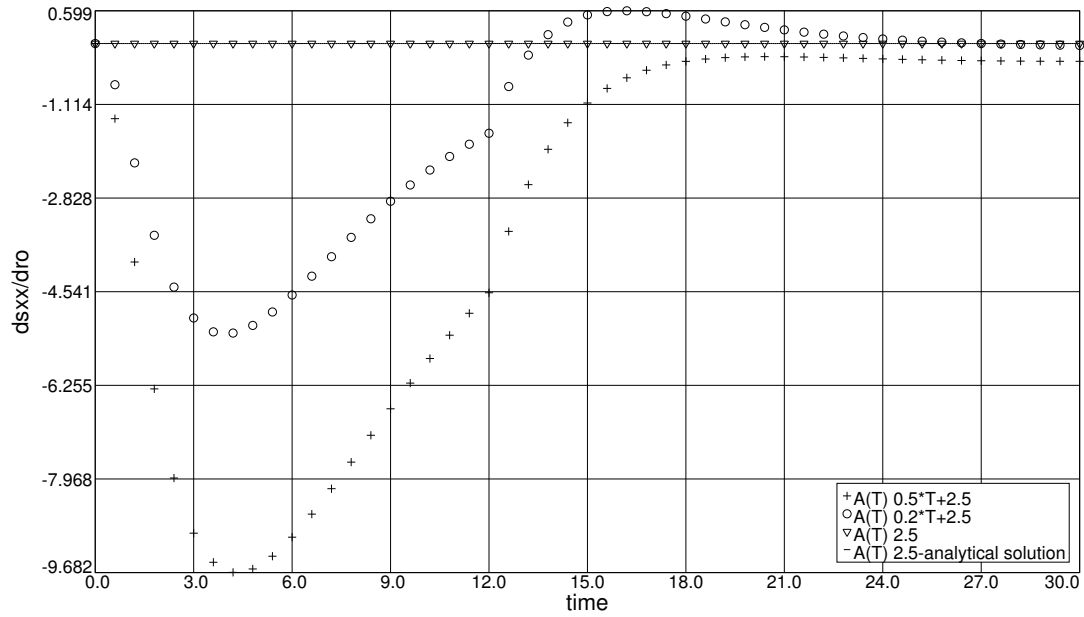
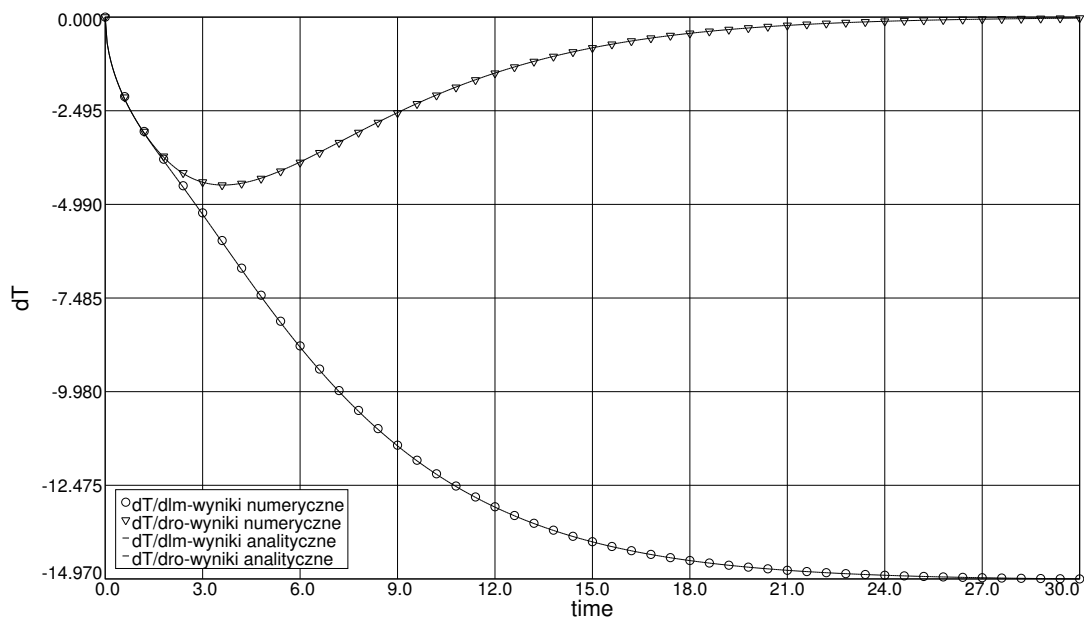
Rysunek 4.4: Niestacjonarny przepływ ciepła. Zmiana temperatury w czasie dla niestacjonarnego problemu przepływu ciepła



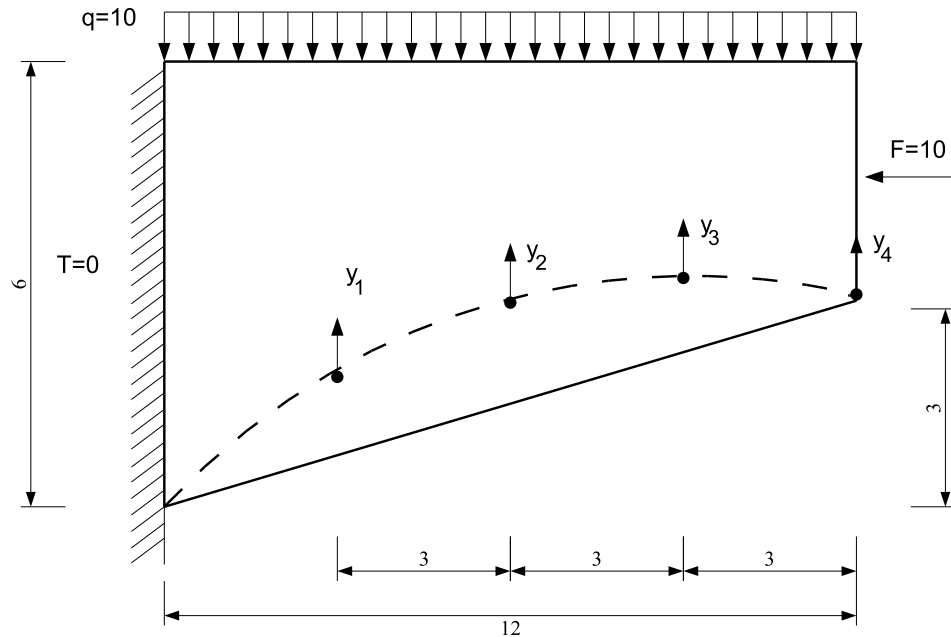
Rysunek 4.5: Wrażliwość naprężenia normalnego na końcu pręta na E^0



Rysunek 4.6: Wrażliwość naprężenia normalnego na końcu pręta na E^∞

Rysunek 4.7: Wrażliwość naprężenia normalnego na końcu pręta ρ Rysunek 4.8: Wrażliwość temperatury na stałe materiałowe przepływu ciepła λ i ρ

4.7 Przykład numeryczny - optymalizacja kształtu wspornika lepkosprężystego



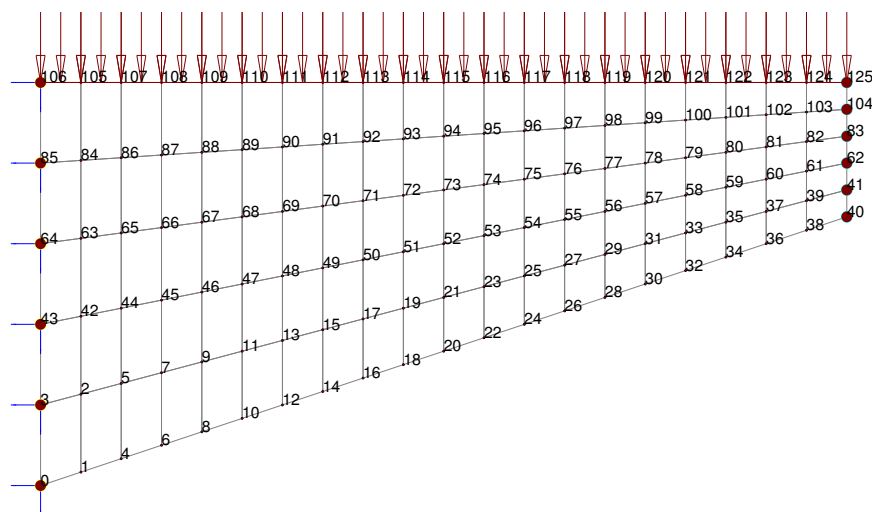
Rysunek 4.9: Schemat statyczny wspornika

Problem optymalizacji kształtu sformułowany jest następująco:

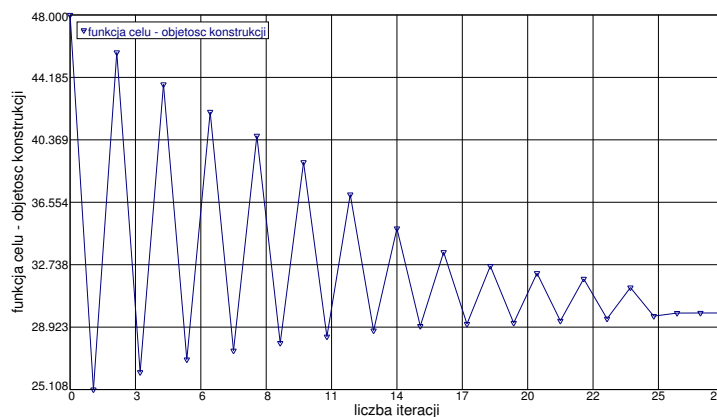
$$\begin{cases} \min V(\mathbf{x}), \\ g_i(\mathbf{x}) = \sigma_M^i|_{\tau=30s} + 150 \leq 0, \quad i = 0, 8, 12, 14, 28, 30, 32, 34, 106, \end{cases} \quad (4.60)$$

gdzie σ_M^i to naprężenie zredukowane Hubera-Misesa w i -tym węźle konstrukcji. Przyjmujemy cztery zmienne projektowe opisujące kształt konstrukcji $\mathbf{x} = \{y_1, y_2, y_3, y_4\}$. Schemat statyczny optymalizowanego wspornika przedstawiony jest na Rys. 4.9. Dolna krawędź wspornika jest opisana wielomianem czwartego stopnia jednoznacznie wyznaczonym przez pięć punktów. Zakres zmienności poszczególnych zmiennych projektowych jest następujący: $y_1 \in [1, 3.5]$, $y_2 \in [2, 4.5]$, $y_3 \in [3, 5]$, $y_4 \in [3, 5.5]$. Sformułowanie problemu (4.60) zawiera 9 ograniczeń. Są to ograniczenia wartości naprężeń zredukowanych Hubera-Misesa, obliczonych na końcu analizy pełzania czyli dla czasu $\tau = 30s$. Jako graniczną, dopuszczalną wartość wartości naprężenia przyjęto 150. Pozostałe dane zadania: materiał termolepkosprężysty, Stałe materiałowe: $E^0 = 500$, $E^\infty = 100$, $\rho = 1$, $\nu = 0.3$, $\kappa = \frac{\lambda}{\rho c} = 1$, wartość strumienia ciepła na swobodnym końcu pręta wspornika $\hat{q} = 1$. Na utwierdzonym końcu pręta zadano warunki brzegowe dla temperatury $T = 0$. Przyjęto funkcję zależności termicznej $A(\theta)$ z równania Eq. 3.18 w następującej postaci $A(\theta) = 0.5 \cdot T + 2.5$. Rozpatrywany przedział czasu wynosi $0 \div 30s$. W tym teście zastosowano 100 równych kroków czasowych. Na Rys. 4.10 przedstawiono dyskretyzację skończenie-elementową zawierającą 100 elementów czterowęzłowych typu Lagrange, z liniowymi funkcjami. Rys. 4.11 obrazuje zbieżność procesu optymalizacji. Po 28 iteracjach osiągnięta została wartość minimalna

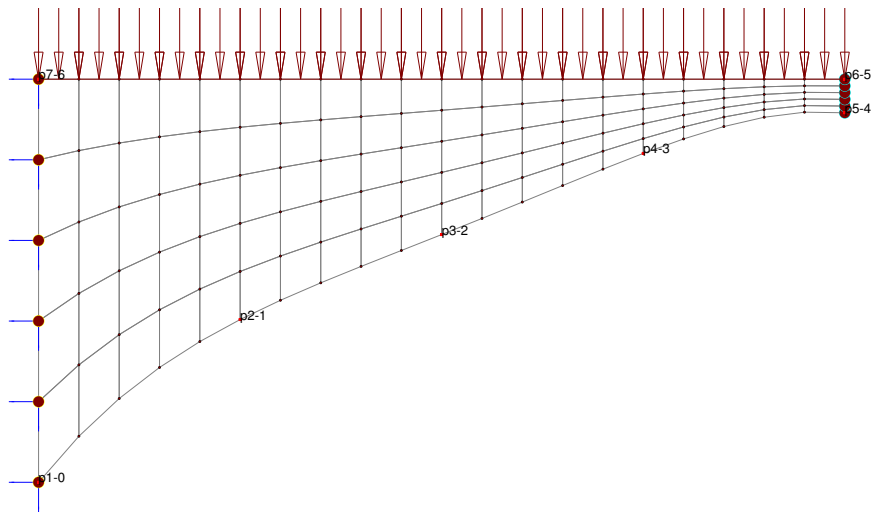
objętości. Początkowa objętość wynosiła 48 po zoptymalizowaniu objętość wynosi 29.8 co stanowi 62% objętości początkowej. Wartości zmiennych projektowych dla konstrukcji optymalnej wynoszą $y_1 = 2.42$, $y_2 = 3.69$, $y_3 = 4.89$, $y_4 = 5.5$. Na Rys. 4.12 przedstawiony został kształt optymalny wspornika. Mapę warstwicową naprężeń Hubera-Misesa przedstawiono na Rys. 4.13. Widać na nim spełnienie ograniczeń zadania.



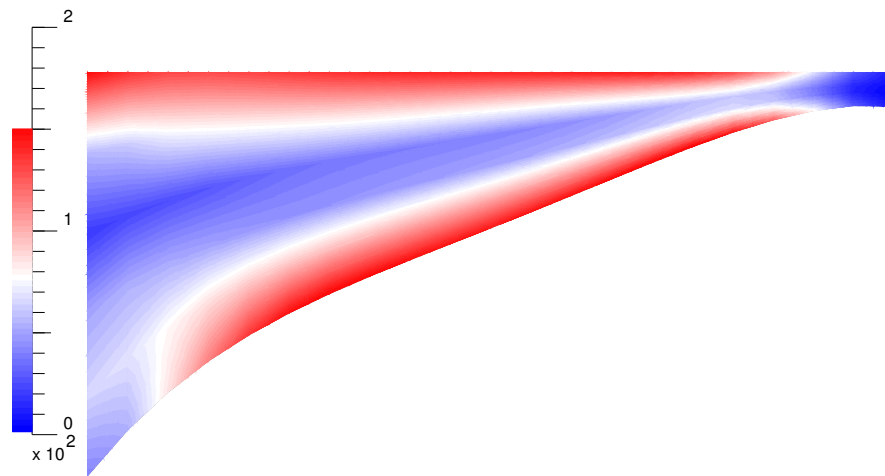
Rysunek 4.10: Dyskretyzacja MES. Zastosowano 100 elementów czterowęzłowych typu Lagrange



Rysunek 4.11: Funkcja celu (objętość wspornika) w procesie optymalizacji



Rysunek 4.12: Optymalny kształt wspornika



Rysunek 4.13: Mapa naprężeń Hubera-Misesa

Ocena niezawodności konstrukcji

5.1 Informacje wstępne

Problem niezawodności konstrukcji sprowadza się do wyznaczenia prawdopodobieństwa awarii. Wiele parametrów projektowych konstrukcji traktowanych jest często deterministycznie. Są to na przykład parametry materiałowe, wartość obciążenia czy parametry opisujące kształt konstrukcji. W rzeczywistości parametry te mają zmienny charakter i uwzględniając tę zmienność otrzymujemy w rezultacie również pewien rozkład wyników analizy MES takich, jak przemieszczenia czy naprężenia. Może się okazać, że pewne niekorzystne wartości zmiennych projektowych mogą powodować niedopuszczalne wartości przemieszczeń, naprężeń bądź ich funkcji (np. naprężeń Hubera-Misesa). Przyjmowanie przez założoną przez projektanta funkcję wartości wynikowych (zwaną dalej funkcją graniczną) wartości uznanych za niedopuszczalne nazywamy umownie awarią konstrukcji, choć nie musi ona oznaczać dosłownego zniszczenia konstrukcji. Wartość liczbowa określającą niezawodność, zwaną inaczej trwałością konstrukcji R , można wyrazić wzorem

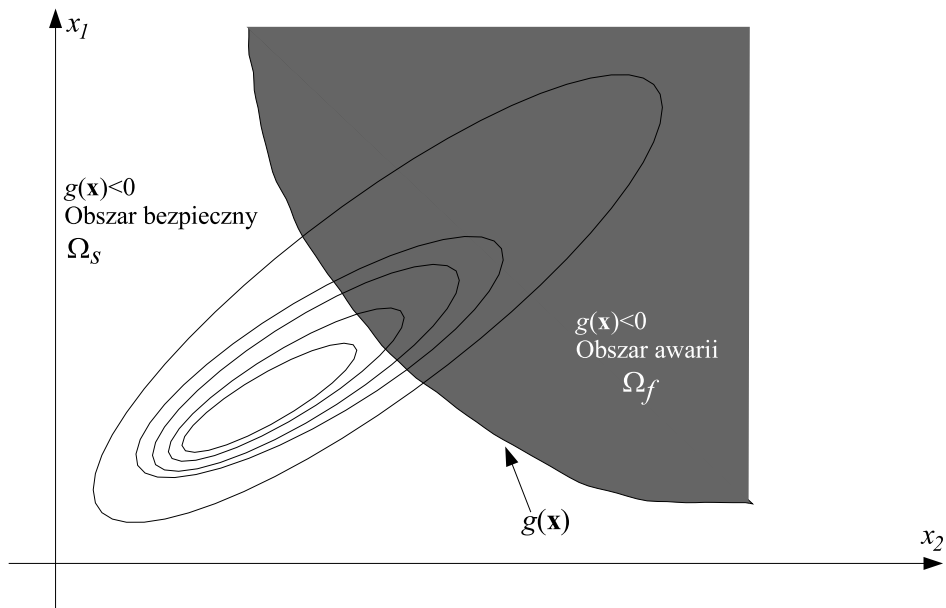
$$R = 1 - P_f, \quad (5.1)$$

gdzie P_f jest prawdopodobieństwem awarii.

Oznaczmy wektor zmiennych losowych jako $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, funkcję graniczną określającą granicę między obszarem awarii i obszarem bezpiecznym oznaczmy jako $g(\mathbf{x})$. W obszarze awarii Ω_f funkcja graniczna przyjmuje wartości ujemne, na granicy obszarów równe zero a w obszarze bezpiecznym jest większa od zera (Rys. 5.1). Funkcja charakterystyczna obszaru awarii przyjmująca wartości niezerowe na Ω_f ma postać

$$I_g(\mathbf{x}) = \begin{cases} 1 : g(\mathbf{x}) \leq 0 \\ 0 : g(\mathbf{x}) > 0 \end{cases} \quad (5.2)$$

Prawdopodobieństwo awarii można obliczyć całkując funkcję łącznej gęstości prawdopodobieństwa $f_{\mathbf{x}}(\mathbf{x})$ zmiennych losowych po obszarze awarii, czyli tam, gdzie $g(\mathbf{x}) < 0$ w/g



Rysunek 5.1: Funkcja graniczna w przestrzeni zmiennych projektowych

wzoru:

$$P_f = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} I_g(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_n. \quad (5.3)$$

Celem projektanta jest stworzenie konstrukcji trwałych, zatem cechować je będzie niskie prawdopodobieństwo awarii. W praktyce wynosi ono zwykle $10^{-4} \div 10^{-7}$, więc wygodnie jest posługiwać się *wskaźnikiem niezawodności* β . Wskaźnik niezawodności może być zdefiniowany następująco:

$$\beta = -\Phi^{-1}(P_f), \quad (5.4)$$

gdzie funkcja Φ jest funkcją odwrotną dystrybuanty rozkładu normalnego. Wskaźnik niezawodności dla w/w przedziału prawdopodobieństw awarii wynosi $3.7 \div 5$.

Bezpośrednie obliczanie prawdopodobieństwa awarii ze wzoru (5.3) jest możliwe tylko dla trywialnych, akademickich przypadków. W praktyce inżynierskiej opracowano szereg metod pozwalających bardziej efektywnie szacować prawdopodobieństwo awarii konstrukcji.

5.2 Charakterystyka metod szacowania niezawodności

5.2.1 Metoda Monte Carlo

Jedną z najprostszych metod obliczania prawdopodobieństwa awarii jest metoda Monte Carlo. Polega ona na wygenerowaniu N realizacji zbioru zmiennych losowych \mathbf{x} . Dla każdej realizacji sprawdza się warunek (5.2) i zlicza liczbę realizacji powodujących awarię.

Prawdopodobieństwo awarii oblicza się korzystając z estymatora wartości średniej

$$P_f = \frac{1}{N} \sum_{i=1}^N I_g(\mathbf{x}_i). \quad (5.5)$$

Metoda ta jest uniwersalna i łatwa do oprogramowania ale posiada też pewne wady. Podstawową jej wadą jest to, że jest kosztowna numerycznie. Dla w miarę dokładnego oszacowania prawdopodobieństwa awarii trzeba przetestować bardzo dużą liczbę realizacji zmiennych losowych. Zwłaszcza w konstrukcjach inżynierskich, dla których, jak wspomniano przeciętnie prawdopodobieństwo awarii nie jest większe zwykle niż 10^{-4} . Dla poprawnego szacowania tego rzędu prawdopodobieństwa ze wzoru (5.5) potrzeba około miliona realizacji zmiennych losowych. Na stosunkowo proste zadanie, którego analiza MES trwa 0.1s potrzeba około 28 godzin dla przetestowania miliona realizacji zmiennej losowej. Czasem jednak pewne własności funkcji granicznej sprawiają, że zastosowanie innych metod staje się niemożliwe. Funkcja graniczna może być zaszumiona lub silnie nieliniowa albo też nie znamy jej własności. W takich przypadkach możemy skorzystać z metody Monte Carlo. Metoda ta może być również stosowana do testowania poprawności innych, bardziej wydajnych metod. Istnieją ulepszenia metody Monte Carlo i w związku z dynamicznym wzrostem mocy obliczeniowych, łatwością implementacji i wszechstronnością zastosowania pojawiają się implementacje tej metody w komercyjnych pakietach do analizy stochastycznej.

5.2.2 First Order Reliability Method (FORM)

W tej metodzie zmienne losowe transformowane są do standardowej przestrzeni gaussowskiej U . W tej przestrzeni prawdopodobieństwo awarii wyrażone jest wzorem

$$P_f = \int_{G(u) \leq 0} \phi(u) du, \quad (5.6)$$

gdzie $\phi(u)$ jest funkcją gęstości prawdopodobieństwa standardowego rozkładu normalnego. Oszacowanie niezawodności tą metodą polega na przybliżeniu hiperpowierzchni granicznej $g(\mathbf{x})$ w przestrzeni gaussowskiej U hiperpłaszczyzną styczną do $g(\mathbf{x})$, w pewnym określonym punkcie u^* (Rys. 5.2). W praktyce uproszczenie polega na tym, że całkowanie po obszarze awarii (5.6) ograniczone jest teraz hiperpłaszczyzną, a nie hiperpowierzchnią, co znacznie upraszcza wyznaczenie prawdopodobieństwa. Takie przybliżenie daje użyteczne oszacowanie prawdopodobieństwa awarii w pewnych określonych warunkach. Zakłada się, że funkcja graniczna nie jest silnie nieliniowa. Założenie to jest prawdziwe dla szerokiej klasy konstrukcji inżynierskich. Punkt, w którym przybliżamy powierzchnię graniczną, jest najbliższym punktem środka układu przestrzeni U i jednocześnie punktem, w którym normalna do krzywej granicznej przechodzi przez środek układu przestrzeni standardowej U . Z własności przestrzeni standardowej wynika, że w tym punkcie jest największa wartość funkcji gęstości prawdopodobieństwa oraz wartość ta maleje wykładniczo, wraz ze zwiększaniem odległości od środka układu. Oznacza to, że otoczenie punktu u^* odgrywa największą rolę w szacowaniu prawdopodobieństwa ze wzoru (5.6). Odległość tego punktu od początku układu jest wskaźnikiem niezawodności β . Wyznaczenie punktu u^* jest najbardziej czasochłonną częścią algorytmu FORM. Do

znalezienia tego punktu stosowane są algorytmy gradientowe optymalizacji minimalizujące odległość punktu projektowego na krzywej od środka układu przestrzeni U . Równanie hiperpowierzchni stycznej do powierzchni granicznej w punkcie u^* można zapisać wzorem

$$\bar{g}(u) = -\alpha^T u + \beta, \quad (5.7)$$

gdzie α jest macierzą gradientów powierzchni granicznej w punkcie u^*

$$\alpha = - \frac{\nabla \bar{g}(u)}{\|\nabla \bar{g}(u)\|} \Big|_{u=u^*} \quad (5.8)$$

Wykorzystując równanie stycznej hiperpłaszczyzny (5.7) w punkcie u^* , wskaźnik niezawodności może być zatem wyrażony następująco

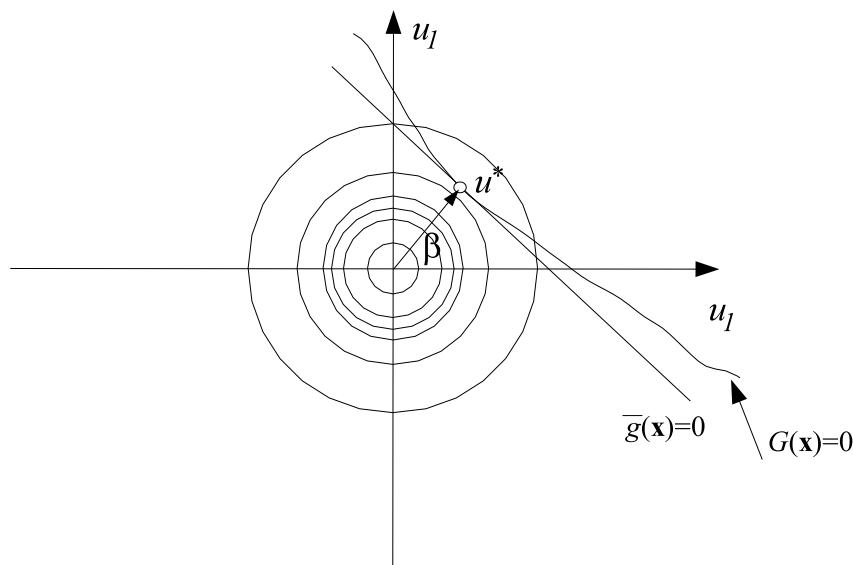
$$\beta = \alpha^T u^*. \quad (5.9)$$

W standardowej przestrzeni gaussowskiej prawdopodobieństwo zdarzenia znajdującego się w podprzestrzeni oddzielonej hiperpłaszczyzną jest równe dystrybucie standardowego rozkładu normalnego z odległością tej płaszczyzny od środka układu jako parametrem dystrybuanty. Zatem w przypadku FORM prawdopodobieństwo awarii wyraża się wzorem

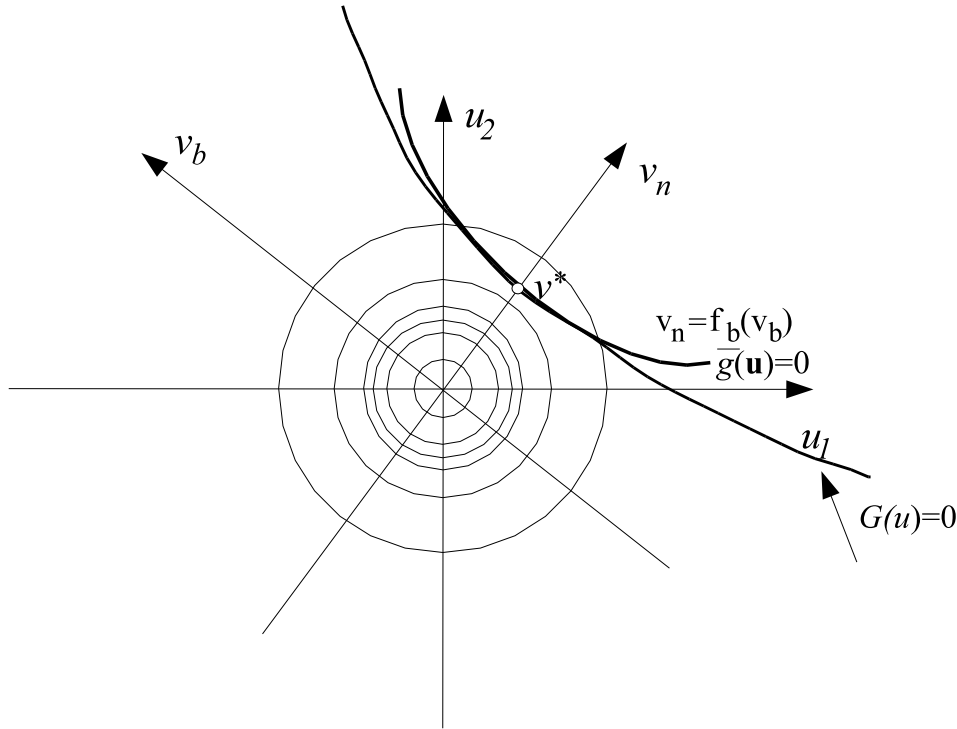
$$P_f = \Phi(-\beta) \quad (5.10)$$

5.2.3 Second Order Reliability Method (SORM)

W niektórych przypadkach liniowe przybliżenie funkcji granicznej może okazać się niewystarczające. Co więcej czasami funkcja graniczna w przestrzeni oryginalnej jest liniowa, a po przetransformowaniu jej do standardowej przestrzeni gaussowskiej staje się funkcją nieliniową. W metodzie SORM powierzchnię graniczną przybliża się powierzchnią drugiego



Rysunek 5.2: FORM



Rysunek 5.3: Funkcja graniczna przybliżona paraboloidą w metodzie SORM

rzędu czyli paraboloidą zawierającą punkt projektowy u^* wyznaczany tak jak w metodzie FORM. Istnieją dwa sposoby wyznaczania paraboloidy przybliżającej powierzchnię graniczną. Pierwszy krzywiznowy (ang. *curvature-fitted*) polegający na wyznaczeniu paraboloidy w oparciu o krzywizny funkcji granicznej w punkcie projektowym u^* . Paraboloida konstruowana jest tu w nowym układzie współrzędnych $v = \{v_1, v_2, \dots, v_n\}$, w którym ostatnia oś zawiera punkt projektowy u^* (Rys. 5.3) w nowym układzie współrzędnych, mający zatem współrzędne $v^* = \{0, 0, \dots, \beta\}$.

Wyrażając funkcję graniczną, jako wartość ostatniej współrzędnej v_n w funkcji pozostałych \mathbf{v}_b , można zapisać

$$v_n = f_g(\mathbf{v}_b). \quad (5.11)$$

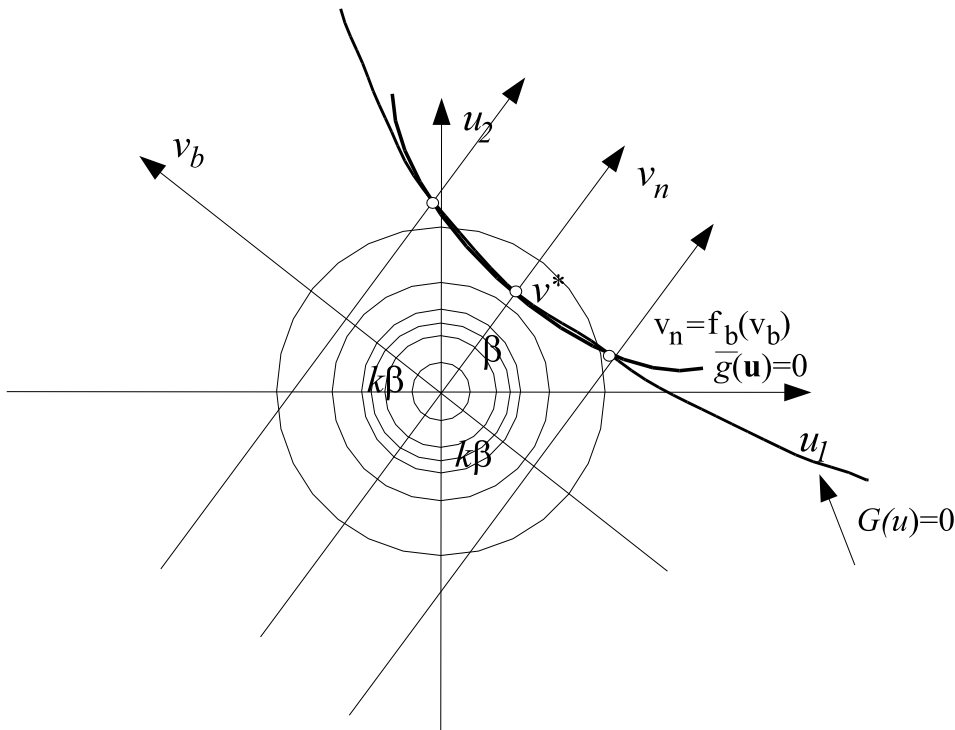
powierzchnia graniczna wyrażona jest następująco:

$$g(\mathbf{v}) = -f_g(\mathbf{v}_b) - v_n. \quad (5.12)$$

Paraboloidę w tym przypadku opisuje relacja

$$f_g(\mathbf{v}_b) = \beta + \frac{1}{2} \sum_{k=1}^{n-1} \kappa_k v_k^2, \quad (5.13)$$

gdzie κ_k to krzywizny główne funkcji $f_g(\mathbf{v}_b)$ (jest ich $n-1$), czyli wartości własne hesjanu w punkcie v^* powierzchni granicznej w punkcie f_g . Oprócz transformacji z układu u do układu v trzeba rozwiązać problem własny hesjanu rzędu $n-1$, co może być czasochłonne w przypadku dużej liczby zmiennych projektowych. Prawdopodobieństwo awarii SORM



Rysunek 5.4: Funkcja graniczna przybliżona paraboloidą poprowadzoną przez trzy punkty

szacujemy ze wzoru wyrażającego całkę po obszarze awarii przybliżonego paraboloidą Δ_{fs} , czyli prawdopodobieństwo zdarzenia $v_n > f_g(\mathbf{v}_b)$

$$P_f = \int_{\Delta_{fs}} \phi(v_n) \phi_{n-1}(\mathbf{v}_b) dv = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \Phi(-f_g(\mathbf{v}_b)) \phi_{n-1}(\mathbf{v}_b) d\mathbf{v}_b, \quad (5.14)$$

$$P_f = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \Phi\left(-\beta - \frac{1}{2} \sum_{k=1}^{n-1} \kappa_k v_k^2\right) \phi_{n-1}(\mathbf{v}_b) d\mathbf{v}_b. \quad (5.15)$$

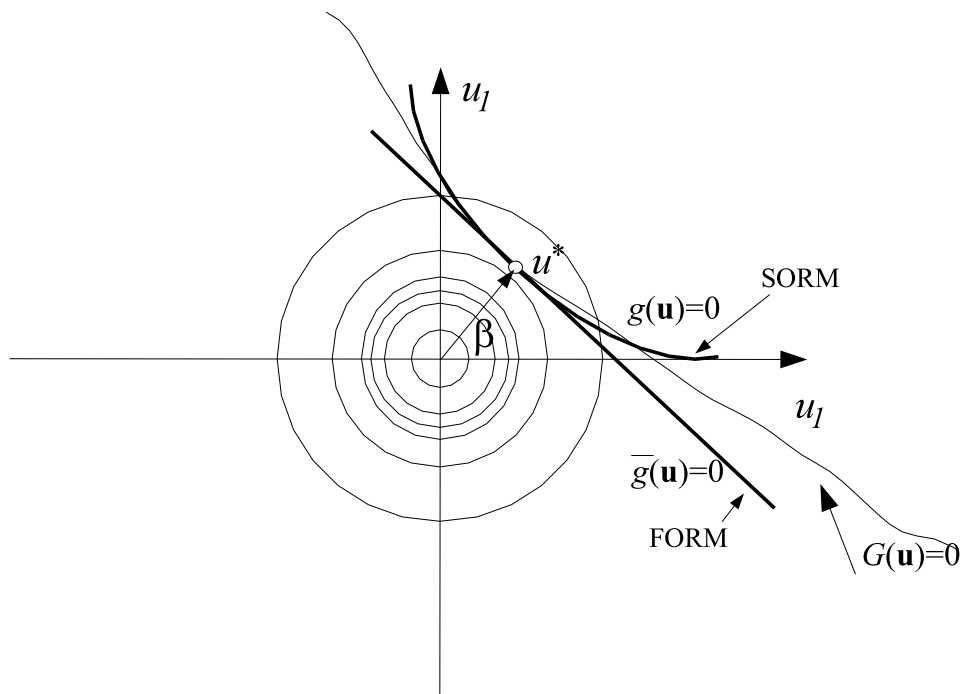
Ostatecznie prawdopodobieństwo awarii wyraża się wzorem

$$P_f \equiv P_f^{SORM} = \Phi(-\beta) \prod_{k=1}^{n-1} [1 + \kappa_k \psi(-\beta)]^{-\frac{1}{2}}. \quad (5.16)$$

Wskaźnik niezawodności metody SORM wynosi

$$\beta^{SORM} = -\Phi^{-1}(P_f^{SORM}). \quad (5.17)$$

Drugi sposób budowy aproksymującej funkcję graniczną paraboloidy, zwany punktowym, (ang. *point-fitted*) polega na napisaniu równania paraboloidy (5.13) przechodzącej przez trzy punkty (Rys. 5.4). Zaletą tej koncepcji jest to, że nie trzeba mieć hesjanu funkcji granicznej w punkcie projektowym v^* . Nie jest zatem konieczne rozwiązywanie problemu



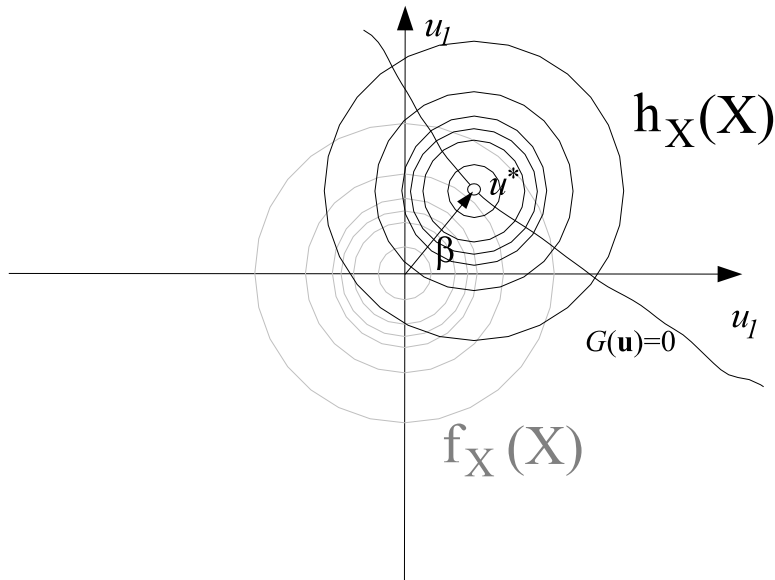
Rysunek 5.5: Porównanie metod FORM i SORM

własnego dla hesjanu, szczególnie kosztownego w przypadku dużej liczby zmiennych projektowych. Dodatkową zaletą tego podejścia jest możliwość szacowania prawdopodobieństwa awarii dla problemów, w których funkcja graniczna jest zaszumiona. Dla zaszumionych funkcji oparcie wyznaczania paraboloidy o wartość hesjanu w punkcie v^* może prowadzić do błędnych wyników.

Metodą SORM otrzymuje się lepsze przybliżenie prawdopodobieństwa awarii niż metodą FORM (Rys. 5.5). Metoda ta jednak jest bardziej czasochłonna co może stwarzać spore problemy w przypadku analizy dużych problemów inżynierskich. Obliczanie funkcji granicznej danej niejawnie wiąże się z wykonaniem analizy mes. Obliczenie hesjanu funkcji granicznej wykonywane jest metodą różnicową, z racji braku możliwości liczenia wrażliwości drugiego rzędu. W przypadku optymalizacji z ograniczeniami niezawodnościowymi wyznaczanie prawdopodobieństwa awarii powtarzane jest wielokrotnie. W przeciwieństwie do FORM, w metodzie SORM znacznie trudniej też wyznacza się wrażliwość wskaźnika niezawodności β^{SORM} , stąd mimo lepszego oszacowania wskaźnika niezawodności przez SORM, obie metody są szeroko stosowane w praktyce inżynierskiej, a w przypadku optymalizacji niezawodnościowej stosuje się głównie FORM.

5.2.4 Importance sampling

Metoda analizy niezawodnościowej *importance sampling* łączy metodę poszukiwania punktu projektowego z przeszukiwaniem losowym, stosowanym w metodzie Monte Carlo. Dzięki takiemu połączeniu potrzebna jest mniejsza liczba realizacji losowych do oszacowania prawdopodobieństwa awarii konstrukcji niż w tradycyjnej metodzie Monte Carlo, gdyż realizacje losowe skoncentrowane są w regionie awarii. Takie przesunięcie koncentracji próbkowania w region awarii opiera się na poniższej zależności definiującej wartość



Rysunek 5.6: Importance sampling

średnią.

$$\begin{aligned}
 E_{\mathbf{X}}(I_{\Omega_f}(\mathbf{X})) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} I_{\Omega_f}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_n. & (5.18) \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} I_{\Omega_f}(\mathbf{Y}) \frac{f_{\mathbf{X}}(\mathbf{Y})}{h_{\mathbf{Y}}(\mathbf{Y})} h_{\mathbf{Y}}(\mathbf{Y}) d\mathbf{y}_1 d\mathbf{y}_2 \dots d\mathbf{y}_n = E_{\mathbf{Y}} \left[I_{\Omega_f}(\mathbf{Y}) \frac{f_{\mathbf{X}}(\mathbf{Y})}{h_{\mathbf{Y}}(\mathbf{Y})} \right]. & (5.19)
 \end{aligned}$$

Powyższe zależności pozwalają obliczać wartość średnią, a co za tym idzie prawdopodobieństwo awarii w innej niż oryginalna przestrzeni \mathbf{Y} i stosując w niej inną funkcję gęstości prawdopodobieństwa $h_{\mathbf{Y}}(\mathbf{Y})$. Oszacowanie prawdopodobieństwa ma zatem postać:

$$P_f = E_{\mathbf{X}}[I_g(\mathbf{X})] = E_{\mathbf{Y}} \left[I_g(\mathbf{Y}) \frac{f_{\mathbf{X}}(\mathbf{Y})}{h_{\mathbf{Y}}(\mathbf{Y})} \right]. \quad (5.20)$$

W standardowej przestrzeni gaussowskiej powyższy wzór można zapisać następująco

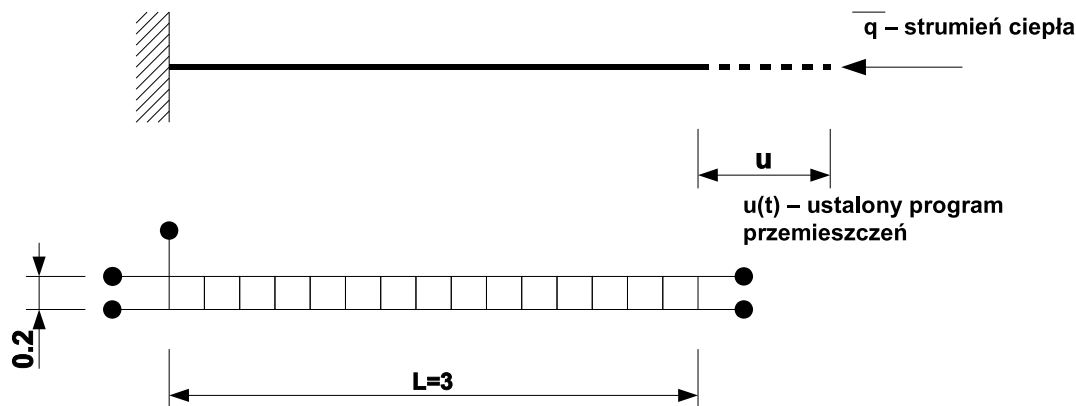
$$P_f = E_{\mathbf{Y}}[I_g(\mathbf{U})] = E_{\mathbf{Y}} \left[I_g(\mathbf{Y}) \frac{\phi(\mathbf{Y}, 0, \mathbf{I})}{h_{\mathbf{Y}}(\mathbf{Y})} \right], \quad (5.21)$$

gdzie $I_g(\mathbf{U})$ to funkcja charakterystyczna obszaru awarii określona wzorem (5.2), funkcja $\phi(u)$ to funkcja gęstości wielowymiarowego standardowego gaussowskiego rozkładu prawdopodobieństwa. Odpowiedni dobór funkcji gęstości prawdopodobieństwa decyduje o efektywności estymatora. Ważne jest, aby koncentracja gęstości $h_{\mathbf{Y}}(\mathbf{Y})$, a tym samym próbkowania losowego następowała w otoczeniu punktu projektowego, w którym koncentruje się rozkład funkcji prawdopodobieństwa awarii. To w dużej mierze sprawia,

że potrzebna jest znacznie mniejsza liczba próbek do oszacowania prawdopodobieństwa awarii.

Istnieje duża liczba odmian metody importance sampling różniących się sposobem doboru funkcji $h_{\mathbf{Y}}(\mathbf{Y})$. Często stosuje się wielowymiarową funkcję rozkładu Gaussa, co zapewnia dobrą efektywność tej metody (tj. o kilku tysięcy próbek). Najlepsze jednak rezultaty uzyskuje się uwzględniając przy doborze funkcji $h_{\mathbf{Y}}(\mathbf{Y})$ informacje o postaci funkcji granicznej. Można wtedy osiągnąć efektywność pozwalającą przybliżyć rozwiązanie za pomocą tylko 100 próbek.

5.3 Przykład numeryczny - analiza niezawodności pręta termolepkosprężystego

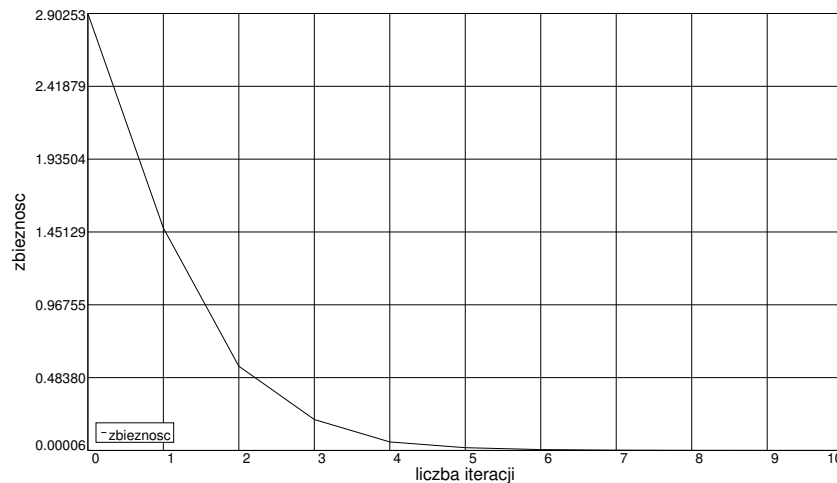


Rysunek 5.7: Schemat pręta. Model pręta składający się z 15 czterowęzłowych elementów typu Lagrange

Poddany zostanie analizie niezawodnościowej pręt w stanie jednoosiowego rozciągania (Rys. 5.7). Przykład tego pręta był już zamieszczony poprzednio w punkcie (p.4.6), gdzie przedstawiono rezultaty analizy wrażliwości. W tym przykładzie porównamy cztery metody analizy niezawodnościowej: FORM, SORM, Importance sampling oraz Monte Carlo. Dane dotyczące własności geometryczno-mechanicznych pozostały bez zmian (por. p.4.6). Przyjęto trzy zmienne losowe: E^0 , E^∞ oraz gęstość ρ . Zmienne losowe E^0 , E^∞ mają rozkład normalny. Gęstość ρ przyjmuje jedynie wartości dodatnie dlatego założono, że ma rozkład logarytmiczno-normalny. Charakterystyki rozkładów prezentuje poniższa tabela

zmienna	wartość średnia m	odchylenie standardowe σ
E^0	500	80
E^∞	100	5
ρ	1	0.5

Rozpatrujemy w naszym przykładzie problem relaksacji naprężeń w pręcie po upływie czasu $t = 12s$ przy obciążeniu geometrycznym końca pręta. Funkcja graniczna zdefiniowana jest następująco: $G(X) = \sigma(X) - \sigma_{lf}$, gdzie $\sigma(X)$ jest naprężeniem w pręcie, a $\sigma_{lf} = 35$. Wyniki analizy niezawodnościowej zestawione są w tabeli 5.1. Otrzymano



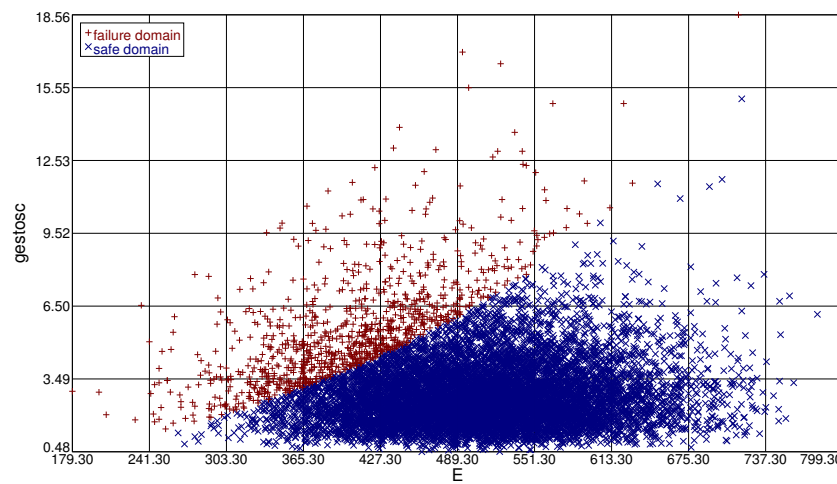
Rysunek 5.8: Zbieżność algorytmu Hasofer-Linda-Rackwitza-Fiesslera poszukiwania punktu projektowego

niemal identyczne rezultaty metodami symulacyjnymi Importance Sampling oraz Monte Carlo. Prawdopodobieństwo obliczone metodą FORM nieco odbiega od pozostałych co

Typ analizy	Liczba analiz MES	czas obliczeń	prawdopodobieństwo awarii	β
FORM	10	0:00:21.5	0.012	2.263
SORM	12	0:00:24.8	0.011	2.292
IS	1010	0:06:12.3	0.011	2.300
MC	10000	0:55:15.2	0.011	2.301

Tablica 5.1: Zestawienie wyników analizy niezawodnościowej

spowodowane jest tym, że prawdopodobieństwo jest stosunkowo duże i założenie liniowego przybliżenia powierzchni granicznej w punkcie projektowym nie prowadzi do dostatecznie dokładnych rezultatów. Znacznie lepsze rezultaty otrzymano metodą SORM wykorzystującą paraboliczną aproksymację powierzchni granicznej. Widać jednak, że metody FORM i SORM są metodami najefektywniejszymi i do oszacowania prawdopodobieństwa awarii wystarczy niewielka liczba analiz. Trzeba mieć na uwadze, że efektywność metody SORM maleje wraz z rosnącą liczbą zmiennych losowych za sprawą algorytmu wyznaczania hesjanu oraz wyznaczania jego wartości własnych hesjanu powierzchni granicznej. Wykres zbieżności stosowanego w FORM i SORM algorytmu poszukiwania punktu projektowego Hasofer-Linda-Rackwitza-Fiesslera pokazuje Rys. 5.8 Efektywność analizy ma duże znaczenie w przypadku zadań o dużej liczbie stopni swobody. Rysunek 5.9 obrazuje próbkowanie w przestrzeni dwóch zmiennych E^0 , ρ . Można na tym rysunku zaobserwować przekrój powierzchni granicznej. Widać na tym przekroju nieliniowy charakter powierzchni granicznej. Przykładowe czasy zamieszczone w tabeli 5.1 uzyskano na dwu-procesorowej maszynie z procesorami PIII 1 GHz.



Rysunek 5.9: Sampling Monte Carlo w przestrzeni dwóch zmiennych E^0 , ρ

Generacja siatki i poprawianie dokładności obliczeń MES

6.1 Wstęp

Rozdział ten poświęcony będzie algorytmom generowania siatek MES stosowanych w programie **VFEM++** oraz metodom poprawiania jakości siatek, zwanym po angielsku *Mesh refinement*.

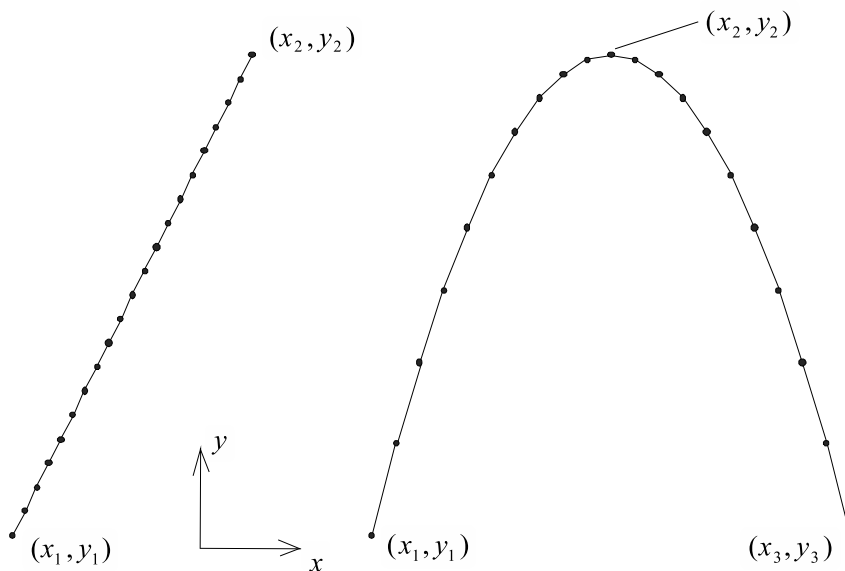
Zasadnicze znaczenie dla rozwiązania zadania metodą elementów skończonych ma dyskretyzacja konstrukcji. Stworzenie odpowiedniej siatki węzłów i elementów ma zasadnicze znaczenie dla uzyskania poprawnych rezultatów obliczeń. Dla pewnych prostych pod względem liczby elementów zadań (w praktyce zwykle prętowych) można brać pod uwagę możliwość ręcznej generacji siatki poprzez podanie explicite listy współrzędnych węzłów i ich wzajemnych połączeń, czyli listy elementów skończonych. Dla większości współczesnych zagadnień zawierających nierzadko dziesiątki czy setki tysięcy elementów skończonych, niezbędne staje się zastosowanie do tworzenia siatek odpowiednich podprogramów zwanych generatorami siatek. Zastosowanie generatorów siatek MES może mieć znaczenie nie tylko ze względu na dużą liczbę elementów skończonych, ale także ze względu na jakość siatki. W metodzie elementów skończonych istnieją kryteria zbieżności, w/g których zagęszczanie siatki prowadzi do uzyskania dokładniejszych wyników. Nadmierne zagęszczanie prowadzi jednak do generowania dużych zadań, których obliczenia są czasochłonne, dlatego ważne jest, aby uzyskiwać siatki dające najdokładniejsze rezultaty dla jak najmniejszej liczby stopni swobody. Odpowiednie rozmieszczanie węzłów i elementów dające najlepsze rezultaty określa się mianem technik adaptacyjnych. Zagadnienie to ma duże znaczenie dla poprawności wyników. Użytkownik programu MES, stosując odpowiedni generator, może zagęszczać siatkę w tych miejscach, w których zadanie tego wymaga (np. koncentracja naprężeń czy strefy zniszczenia). Można to robić jedynie w oparciu o własne doświadczenie. Lepiej jednak nie opierać się na zawodnych przecuciach tylko na obiektywnej mierze błędu. W tym celu stworzone zostały *estymatory*

błędu (ang. *error estimator*) pozwalające na oszacowanie błędu rozwiązania zadania MES. Drugim ważnym elementem procesu tworzenia optymalnej siatki są techniki adaptacyjne. Techniki te to sposoby wykorzystania estymatora błędu do poprawienia jakości siatki. Wartość estymatora w danym punkcie konstrukcji determinuje rozmiar elementu jaki powinna mieć siatka w danym miejscu. Takie działanie sprawi, że siatka będzie zagęszczona w miejscach o dużej wartości błędu i bardziej rzadka tam, gdzie błąd jest mniejszy.

W programie zaimplementowano możliwość ręcznego wprowadzania węzłów i elementów oraz automatyczną generację siatek regularnych oraz triangularyzację.

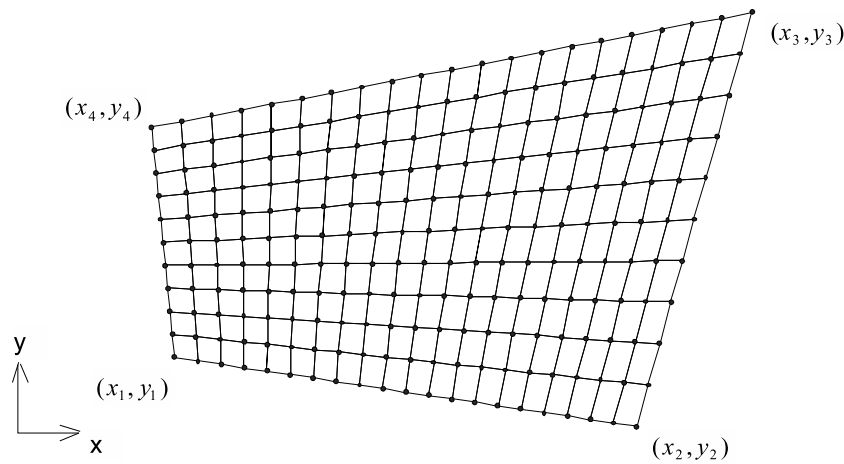
6.2 Siatki regularne

W programie VFEM++ zdefiniowano szereg klas służących do generowania siatek. Pewne szczegóły struktur obiektowych generacji siatek są opisane w rozdziale drugim. Do wygenerowania siatki potrzebne jest zdefiniowanie rodzaju układu współrzędnych, w którym będziemy definiować obszar do siatkowania. Ponadto należy zdefiniować typ siatki, jaki ma być użyty. Dla zadań płaskich mamy do dyspozycji układy współrzędnych kartezjański oraz biegunowy, zaś dla przestrzennych układy: kartezjański, sferyczny i walcowy. W ogólności struktury obiektowe przewidują istnienie wielu typów siatek. Obecnie zaimplementowano dwa rodzaje siatek. Dla elementów jednowymiarowych jest to równomierny podział odcinka na n elementów. Dla elementów powierzchniowych podział prostokątnego obszaru na $n \times m$ elementów powierzchniowych. Elementy jednowymiarowe mogą być generowane na odcinku krzywej zdefiniowanym przez współrzędne końców lub na odcinku krzywej zdefiniowanym przez trzy punkty: współrzędne końca i współrzędne środka odcinka

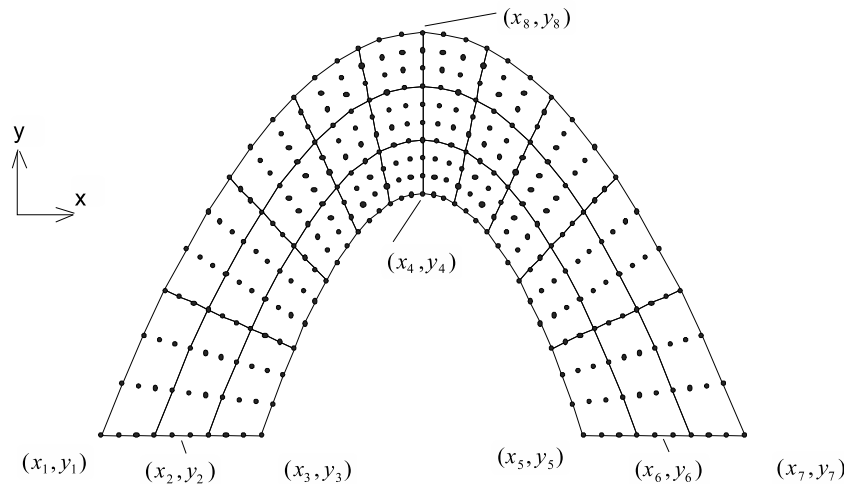


Rysunek 6.1: Przykład siatek elementów jednowymiarowych w układzie kartezjańskim

Do definiowania obszarów dwuwymiarowych lub powierzchni pokrywanych siatkami, używa się aproksymacji **Lagrange4** oraz **Serendip8**. Pozwala ona na zdefiniowanie obszaru przez podanie kilku punktów węzłowych wyznaczających jednoznacznie dany obszar poprzez

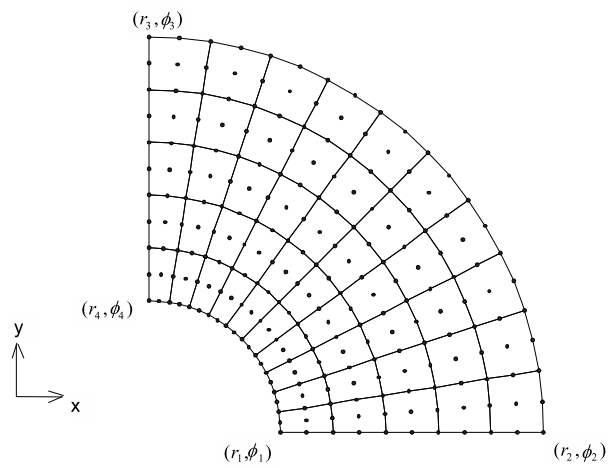


Rysunek 6.2: Siatka na obszarze wyznaczonym przez aproksymację Lagrange4 w układzie kartezjańskim

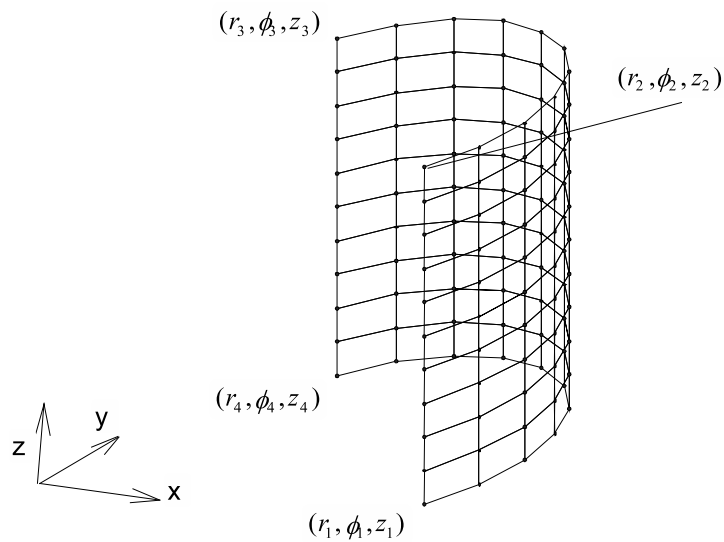


Rysunek 6.3: Aproksymacja Serendip8 użyta do zdefiniowania obszaru generacji siatki w układzie kartezjańskim

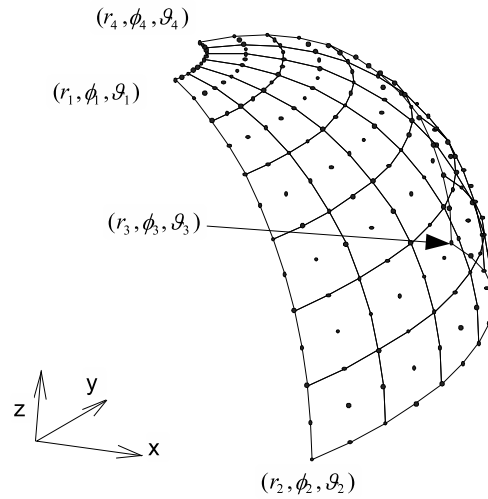
aproksymacje funkcjami kształtu. Poniżej zaprezentowanych zostanie kilka przykładów siatek regularnych. Na rysunku 6.2 pokazano obszar opisany przy pomocy aproksymacji Lagrange4 i definiowany przez cztery punkty. Wygenerowana siatka składa się też z elementów o aproksymacji Lagrange4. Na rysunku 6.3 znajduje się siatka składająca się z elementów Lagrange16, wygenerowana na obszarze opisanym przy pomocy aproksymacji Serendip8, zdefiniowanej przez punkty wierzchołkowe oraz punkty leżące na środkach boków prostokąta. Na Rys. 6.4. pokazano obszar Lagrange4 lecz w układzie biegunowym, pokrytym siatką elementów Lagrange9. Na rysunkach 6.5-6.8 przedstawiono wygenerowane siatki elementów powłokowych na powierzchniach w różnych układach współrzędnych.



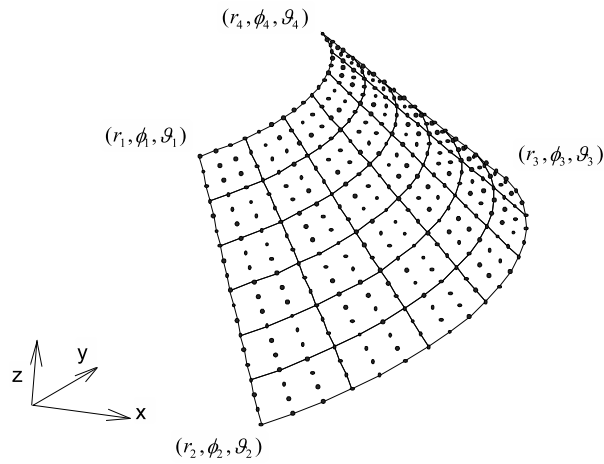
Rysunek 6.4: Obszar definiowany przy pomocy aproksymacji Lagrange4 . Układ biegunowy



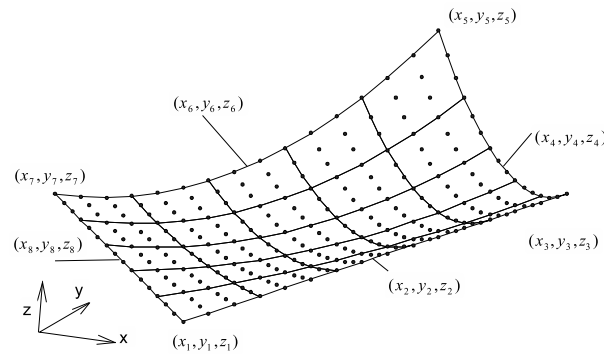
Rysunek 6.5: Obszar definiowany przy pomocy aproksymacji Lagrange4 . Układ walcowy



Rysunek 6.6: Aproksymacja obszaru Lagrange4 w układzie sferycznym. Siatka z elementów Lagrange9



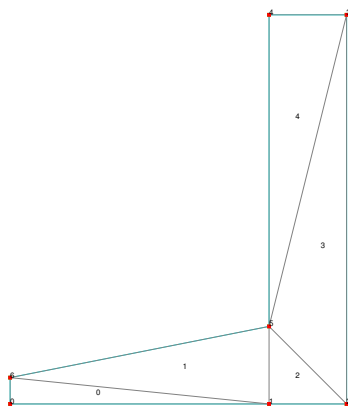
Rysunek 6.7: Powierzchnia stożkowa wygenerowana w układzie sferycznym. Siatka z elementów Lagrange16



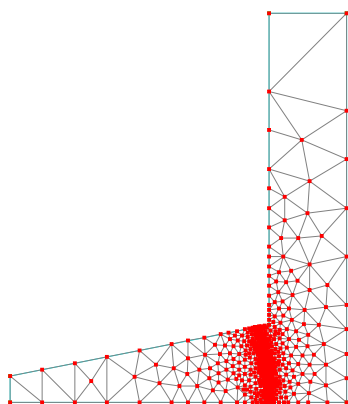
Rysunek 6.8: Obszar generowany przy pomocy aproksymacji §8. Siatka z elementów Lagrange16

6.3 Triangulacja parametryczna Delaunaya

Siatki regularne mogą być stosowane w konstrukcjach o prostych kształtach brzegu. W przypadku bardziej złożonych kształtów trzeba zastosować bardziej złożone algorytmy generacji siatki. Bardzo szeroko stosowanym algorytmem generowania siatki m.in. w metodzie elementów skończonych jest zastosowany w tej pracy algorytm Triangulacji oparty na koncepcji Delaunay'a służący do generowania siatek składających się z trójkątnych elementów skończonych. Oprócz możliwości generowania skomplikowanego kształtu dla analizowanej konstrukcji, drugą istotną cechą generatora siatek jest możliwość generowania siatki zadaną funkcją gęstości węzłów. Pozwoli to na wygenerowane siatki adaptacyjnej, w której stopień zagęszczenia węzłów w danym punkcie konstrukcji zależy od wartości estymatora błędu w tym punkcie. Proces generacji siatki trójkątnej



Rysunek 6.9: Wstępna triangulacja obszaru



Rysunek 6.10: Gotowa siatka trójkątna z lokalnym zagęszczeniem

składa się z kilku etapów:

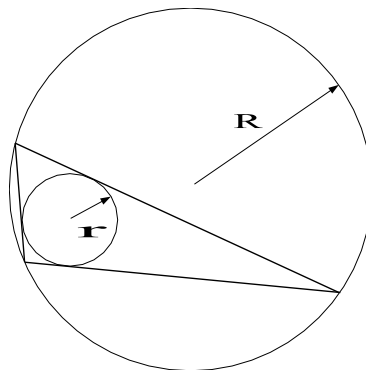
1. Triangularyzacja wstępna. Polega na wstępnym, zgrubnym podziale obszaru na trójkąty. Może być wykonana nawet ręcznie gdyż zwykle nie zawiera dużej liczby trójkątów i jej ręczne stworzenie nie jest specjalnie uciążliwe, Rys. 6.9.
2. Właściwa triangularyzacja (ang. refinement stage) polegająca na kontroli pewnych własności trójkątów, krawędzi oraz funkcji gęstości i podziale na coraz mniejsze elementy.
3. Wygladzanie siatki (Laplacian smoothing), Rys. 6.10.

6.3.1 Triangularyzacja właściwa

Proces podziału trójkątów i krawędzi na mniejsze polega na analizie własności siatki, czyli ocenie jakości trójkątów i krawędzi w siatce. Triangularyzacja rozpoczyna się od utworzenia kolejki takich niewłaściwych trójkątów (ang. *bad triangles*) i krawędzi (ang. *encroached edges*).

Najpierw zajmujemy się własnościami trójkątów i ich podziałem. Najlepszym trójkątem jest trójkąt równoboczny. W konstrukcjach o kształtach nieregularnych nie jest oczywiście możliwe, aby siatka składała się z samych trójkątów równobocznych. Siatka na początku triangularyzacji właściwej składa się z trójkątów o kształtach odbiegających czasem nawet znacznie od trójkąta równobocznego. Stąd kryteria jakości trójkątów mają na celu sprawdzenie stopnia ich podobieństwa do trójkąta równobocznego. Trójkąt nazywamy *trójkątem Delaunaya*, jeśli koło opisane na nim nie zawiera żadnego innego węzła siatki, jak tylko węzły tego trójkąta. Kryterium to wiąże się z następnymi, opisanymi poniżej kryteriami, gdyż w praktyce, im bardziej trójkąt jest zbliżony do trójkąta równobocznego tym mniejsze jest koło na nim opisane i tym mniejsza szansa zawierania się w nim innych węzłów siatki. Innym sposobem oceny jakości trójkąta jest porównanie promienia okręgu wpisanego w trójkąt r z promieniem okręgu opisanego na trójkącie R (Rys. 6.11). Trójkąt uznaje się za dobry, gdy spełnia poniższy warunek:

$$Q = \frac{R}{2r} \leq Q_{min} \quad (6.1)$$



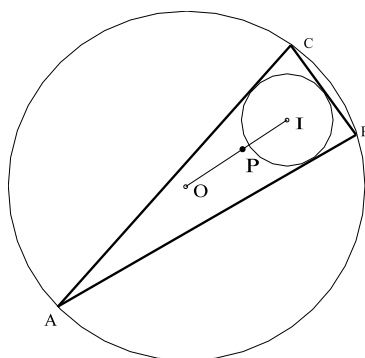
Rysunek 6.11: Ocena jakości trójkąta

Innym kryterium jest sprawdzenie najmniejszej wartości kąta w trójkącie. Stosując to kryterium, otrzymuje się właściwie takie same rezultaty, jak z porównywania promieni

trójkątów wpisanego i opisanego z tym, że mniej obliczeń potrzeba aby wyznaczyć minimalny kąt w trójkącie, niż wymaga wyznaczenie promieni trójkątów opisanego i wpisanego na trójkącie. Z testów szybkości przeprowadzonych VFEM++ wynika, że wyznaczenie promieni wpisanego i opisanego zajmuje trzykrotnie więcej czasu niż minimalnego kąta. Dlatego kryterium minimalnego kąta sprzyja efektywności algorytmu generacji siatki.

Inną istotną cechą elementu jest jego wielkość. Opisane w następnym punkcie oszacowanie błędu przy pomocy estymatora daje w rezultacie informację o tym czy dany element skończony ma właściwe rozmiary. Ta informacja stanowi drugie kryterium decydujące o tym czy proces podziału na mniejsze trójkąty ma być kontynuowany czy nie. Elementy o właściwych kształtach będą mimo to dzielone na mniejsze jeśli, z rozkładu błędu będzie wynikało, że są za duże.

Jeśli już wiemy, że dany trójkąt w siatce musi zostać podzielony na mniejsze procedura podziału przebiega następująco. Najpierw wyznaczamy nowy przyszyły węzeł siatki (*ang. prospective node*). Sposób jego wyznaczenia obrazuje Rys. 6.12. Nowy węzeł siatki



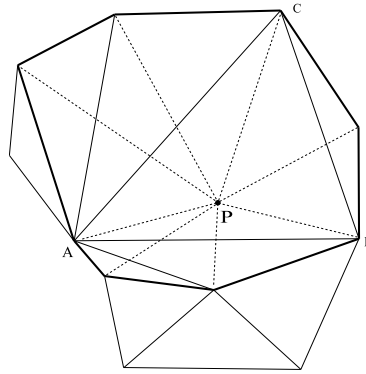
Rysunek 6.12: Sposób wyznaczenia nowego węzła siatki

leży na prostej między środkami okręgu wpisanego i opisanego na okręgu, a jego współrzędne wyraża relacja:

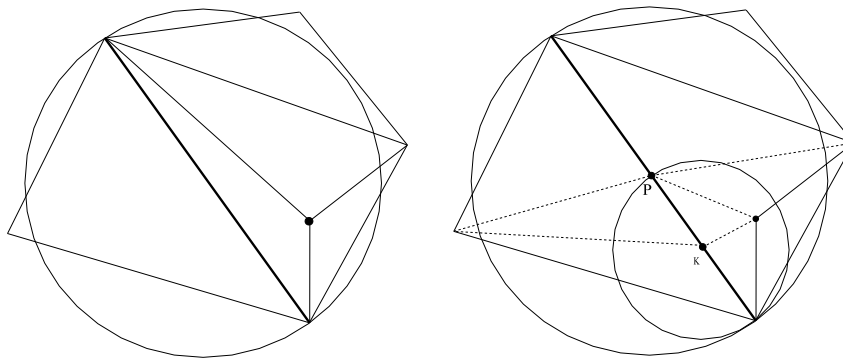
$$\mathbf{X}_P = (1 - \rho)\mathbf{X}_I + \rho\mathbf{X}_O \quad (6.2)$$

gdzie ρ jest wagą z przedziału $(0, 1)$. Więcej szczegółów na temat doboru tej wagi zawiera praca [2]. Mając nowy punkt węzłowy znajdujemy wszystkie sąsiednie trójkąty, których koła opisane na nich, zawierają nowy węzeł P (Rys. 6.13). Granicę tego obszaru pokazano na Rys. 6.13 grubą linią. Następnie retriangularyzujemy siatkę, tworząc nowe trójkąty z poszczególnych odcinków łamanej granicy obszaru i nowego punktu węzłowego P . Stare węzły i elementy leżące w miejscu retriangularyzacji zostają usunięte. Warto odnotować, że ten sposób retriangularyzacji sprzyja spełnianiu przez nowe trójkąty kryterium Delaunaya.

Drugim elementem siatki podlegającym testowaniu i podziałowi są krawędzie. Krawędź będzie podlegała podziałowi, jeśli nie spełnia kryterium Delaunaya lub jeśli jest za długa. Krawędź jest krawędzią Delaunaya, jeśli koło, którego krawędź jest średnicą nie zawiera żadnych innych węzłów siatki, niż końcowe tej krawędzi. W przeciwnym wypadku nowy węzeł siatki tworzony jest w środku krawędzi i siatka jest modyfikowana zgodnie z Rys. 6.14. Powstałe na skutek podziału krawędzie są również sprawdzane, czy spełniają kryterium Delaunaya. Proces ten jest kontynuowany rekurencyjnie aż do spełnienia własności



Rysunek 6.13: Retriangulacja siatki



Rysunek 6.14: Podział krawędzi

Delaunaya przez wszystkie krawędzie, co pokazano na lewym rysunku Rys. 6.14. Na decyzję o podziale krawędzi ma też wpływ jej wielkość. Proces podziału jest kontynuowany, jeśli krawędź nie jest krótsza, niż to wynika z funkcji gęstości węzłów.

6.4 Oszacowanie błędu

Od początku istnienia metody elementów skończonych dokładność obliczeń MES była jednym z ważniejszych zagadnień. Istnieją różne przyczyny powstawania błędów. W statyce liniowej głównym powodem niedokładności obliczeń jest dyskretyzacja MES. Dla problemów nieliniowych, lub problemów zależnych od czasu dodatkowymi błędami zależy również od: przyjętego schematu iteracyjnego, oraz schematu całkowania równania. Nie ma ogólnej teorii szacowania błędów dla szerokiej gamy nieliniowych zagadnień, można jedynie znaleźć takie oszacowania dla pewnych klas problemów. W ostatnich latach powstało szereg prac poświęconych oszacowaniu błędu w modelach zniszczenia [78, 75, 81, 74, 73]. Proponowane tam rozwiązania są jednak bardzo skomplikowane i trudne do zaprogramowania. Jako alternatywę dla złożonych estymatorów błędu stosujemy w modelu zniszczenia stosowany będzie wskaźnik błędu. W obszarach niezniszczonych błąd szacowany będzie za pomocą estymatora Zienkiewicza-Zhu. Oba oszacowania służą do określania błędu spowodowanego dyskretyzacją MES. Dobór właściwej siatki potrzebnej do rozwiązania zadania metodą elementów skończonych nie jest zagadnieniem trywialnym. Kryteria

zbieżności, które zwykle są spełniane przez najczęściej stosowane elementy skończone sprawiają, że zagęszczając siatkę otrzymujemy dokładniejsze rozwiązanie. Zwiększanie liczby stopni swobody powoduje jednak wzrost czasu obliczeń. Potrzebne są metody pozwalające na oszacowanie rozkładu błędu i w konsekwencji stworzenie odpowiedniej siatki zagęszczonej w tych obszarach, gdzie to jest konieczne dla uzyskania dobrych rezultatów. Estymator, oraz wskaźnik błędu są właśnie takimi oszacowaniami błędu, dającym w konsekwencji oszacowanie wielkości elementu pozwalającej na dokonanie zagęszczenia siatki w newralgicznych miejscach konstrukcji.

6.5 Estymator błędu Zienkiewicza-Zhu

Zaprezentowany poniżej estymator błędu Zienkiewicza-Zhu można zastosować dla wielu zagadnień. Może być zastosowany dla różnych typów konstrukcji od statyki liniowej do nawet konstrukcji sprężysto-plastycznych (por. [50]). Dlatego też wartości naprężeń czy odkształceń w poniższych równaniach mogą być rozumiane jako wielkości na kroku czasowym (${}^{t+\Delta t}(\cdot)$). Dalsze rozważania ograniczymy do zagadnień płaskich.

Rozwiązując zadanie metodą elementów skończonych dostajemy rozwiązanie przybliżone. Błąd obliczeń zatem będzie różnicą pomiędzy dokładnym rozwiązaniem, a rozwiązaniem przybliżonym $\mathbf{e}_\sigma = \sigma - \hat{\sigma}$, $\hat{\sigma} = \mathbf{D}\mathbf{B}\mathbf{q}$. Całkowita miara błędu rozwiązania dla obszaru konstrukcji Ω , wyrażona przy pomocy normy energetycznej, będzie zatem miała postać:

$$\|\mathbf{e}\|_\Omega^2 = \int_\Omega \mathbf{e}_\sigma^T \mathbf{D}^{-1} \mathbf{e}_\sigma d\Omega, \quad (6.3)$$

gdzie \mathbf{D} jest macierzą materiałową problemu. Relatywny błąd zdefiniowany jest następująco:

$$\eta = \frac{\|\mathbf{e}\|_\Omega}{\|u\|_\Omega}, \quad (6.4)$$

gdzie $\|u\|_\Omega$, energia odkształcenia zdefiniowana jest następująco:

$$\|u\|_\Omega = \int_\Omega \varepsilon^T \mathbf{D} \varepsilon d\Omega \quad (6.5)$$

Współczesnych bardzo skomplikowanych problemów analizowanych metodą elementów skończonych nie można rozwiązywać analitycznie. Stąd konieczność zastąpienia dokładnego rozwiązania w powyższych rozważaniach inną wielkością. Zienkiewicz i Zhu opracowali estymator błędu, w którym w energetycznej normie błędu zastąpiono dokładne rozwiązanie rozwiązaniem uśrednionym σ^* , zatem całkowity błąd będzie wyrażony wzorem:

$$\overline{\|\mathbf{e}\|_\Omega^2} = \int_\Omega (\sigma^* - \hat{\sigma})^T \mathbf{D}^{-1} (\sigma^* - \hat{\sigma}) d\Omega = \sum_{i=1}^{NE} \int_\Omega (\sigma^* - \hat{\sigma})^T \mathbf{D}^{-1} (\sigma^* - \hat{\sigma}) d\Omega, \quad (6.6)$$

gdzie NE jest liczbą elementów skończonych w zadaniu. Dopuszczalna norma błędu dla pojedynczego elementu wynosi

$$e_a = \eta_t \sqrt{\frac{\|\hat{\mathbf{q}}\|_\Omega^2 + \overline{\|\mathbf{e}\|_\Omega^2}}{NE_e}}. \quad (6.7)$$

Elementowy wskaźnik błędu

$$\xi_i = \frac{\|\mathbf{e}\|_{\Omega_i}}{e_a}. \quad (6.8)$$

Mając obliczone powyższe wielkości, można wyznaczyć nowy rozmiar elementu.

$$h_{new} = \frac{h_{old}}{r_i}, \quad \text{gdzie } r_i = (\xi_i)^{\frac{1}{\mu+1}}. \quad (6.9)$$

Nowa wielkość elementu stanowi funkcję określającą wielkość elementu nowej siatki adaptacyjnej.

6.6 Wskaźnik błędu

Prostą i bardzo wygodną alternatywą dla skomplikowanych estymatorów błędu modeli zniszczenia jest zastosowanie wskaźnika błędu. Podejście to zastosował po raz pierwszy Huerta [77]. Różnica między estymatorem a wskaźnikiem błędu polega na tym, że estymator błędu oparty jest na matematycznych rozważaniach mających na celu oszacować wartość błędu. Wskaźnik zaś, ma jedynie pokazać, w których obszarach konstrukcji błąd jest większy a w których mniejszy. Innymi słowy estymator to oszacowanie ilościowe, a wskaźnik jakościowe. Wielkość elementu skończonego dla poszczególnych obszarów trzeba dobrać na podstawie innych rozważań. Wskaźnik błędu zwykle funkcją zmiennych stanu lub pewnych wielkości związanych z siatką. Dlatego też można go wyznaczyć łatwo i tanim kosztem. W naszym modelu zniszczenia wskaźnikiem błędu będzie wprost wskaźnik zniszczenia d . To proste, i efektywne podejście stosowane jest w wielu pracach (np. [80, 82]). Wyznaczenie wielkości elementu skończonego w obszarze zniszczenia wygląda zatem następująco:

$$h_{new} = h_0 - (h_0 - h_1)d. \quad (6.10)$$

Rozmiar elementu jest liniowo interpolowany pomiędzy wielkościami dla strefy niezniszczonej ($d = 0$) h_0 oraz całkowicie zniszczonej ($d = 1$) h_1 . Wielkość elementu w strefach niezniszczonych może być wyznaczana przy pomocy estymatora ZZ. Trzeba jedynie pamiętać, o tym, żeby wielkość elementu zapewniała właściwą interakcję elementów w nielokalnym zniszczeniu (3.84). W praktyce rozmiar elementu nie powinien być większy niż $1.5R$.

6.7 Techniki adaptacyjne

Istnieje kilka metod poprawiania jakości siatek w oparciu o estymator błędu. Są to:

- Poprawianie jakości typu p (ang. *p-refinement*.) polega ono na zmianie stopnia wielomianów interpolacyjnych funkcji kształtu elementów skończonych.
- Poprawianie jakości typu r (ang. *r-refinement*.) Ten typ poprawienia jakości siatki polega na zmianie współrzędnych węzłów, przy zachowaniu tej samej liczby stopni swobody, celem uzyskania rozkładu gęstości węzłów, podobnego do rozkładu błędu.

- Poprawianie jakości typu h (ang. *h-refinement*.) Ten rodzaj poprawiania siatek realizuje się przez zmianę siatki, lokalne zagęszczanie albo ponowną generację siatki.

W tej pracy wykorzystujemy poprawianie jakości typu h w dwóch wersjach: dla siatek regularnych oraz dla siatek trójkątów. Dla problemów nieliniowych czy zależnych od czasu kiedy to układ równań MES rozwiązywany jest wielokrotnie, dokładności siatki, należy sprawdzać na każdym kroku czasowym. W przypadku, gdy okaże się, że należy zmodyfikować siatkę można wybrać jedną z dwóch dróg postępowania: kontynuowanie analizy, konieczne wtedy jest przeniesienie wszystkich wartości elementowych (wskaźnik zniszczenia, naprężenia lepkie itp.) ze starej siatki na nową. Mapowanie to, można wykonać jedną z trzech metod:

- Metoda najbliższego punktu. Wielkości w danym punkcie elementu (zwykle środka elementu) skończonego nowej siatki są kopiowane z najbliższego takiego punktu na starej siatce. Metoda ta stosowana dla elementów z liniowymi funkcjami kształtu.
- Mapowanie przy pomocy funkcji kształtu. Wielkość w danym punkcie konstrukcji wyliczana jest przez interpolację tej wielkości funkcjami kształtu w tym samym punkcie, na elemencie starej siatki.
- Trzecia metoda polega na wyliczeniu wartości w punkcie konstrukcji jako średniej ważonej wartości w elementach sąsiednich do elementu, wewnątrz którego ten punkt konstrukcji leży na starej siatce.

Druga metoda postępowania restartowaniu analizy od początku za każdym razem, gdy siatka wymaga modyfikacji. Nie ma wówczas potrzeby mapowania wielkości, ale powtórzenie analizy od początku sprawia, że metoda ta jest z reguły mniej efektywna. Mając pewne doświadczenie można jednak czasami przewidzieć na ile niedokładna jest siatka i ile razy w procesie będzie ona wymagała poprawiania. W modelach reologicznych, przy niezmiennych obciążeniach, zdarza się, że siatka wymaga modyfikacji tylko raz na początku analizy. Wówczas warto zastosować tą metodę. Z zastosowaniem tego podejścia można się spotkać w pracach [76, 78].

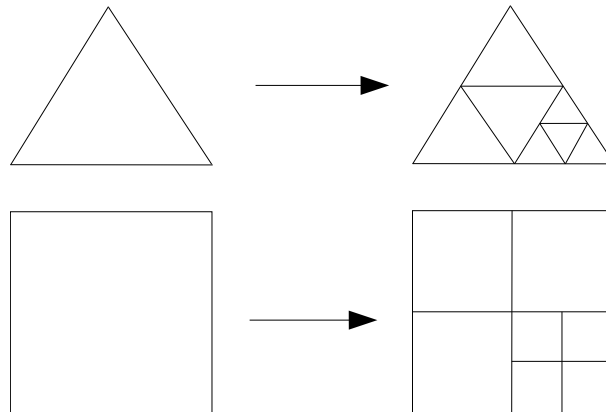
6.8 Poprawianie siatki regularnej przez zagęszczenie

Dla siatek regularnych (takich, jak np Rys. 6.2) poprawianie można wykonywać przez podział elementów na kilka mniejszych, zwanych też podziałem izotropowym. Jest to jedna z najprostszych technik zagęszczania siatek (Rys. 6.15). Podział elementów na mniejsze stwarza jednak pewne trudności. Po podziale elementów niektóre węzły mniejszych elementów przylegają do krawędzi większych elementów. Wymaga to zastosowania aproksymacji związanej (*ang. constrained approximation*), opisaną poniżej w celu zapewnienia ciągłości pola przemieszczeń.

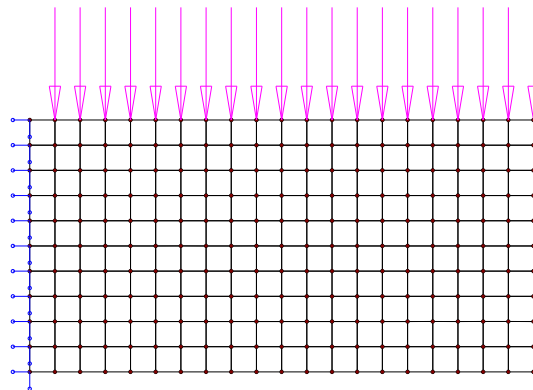
6.8.1 Aproksymacja związana

Na Rys. 6.18 pokazano fragment siatki po wykonaniu zagęszczania siatki prostokątnej, dla której konieczne jest zastosowanie aproksymacji związanej. Konieczność ta wynika z tego, że po zagęszczeniu węzły mogą leżeć na krawędzi elementów skończonych. Węzły

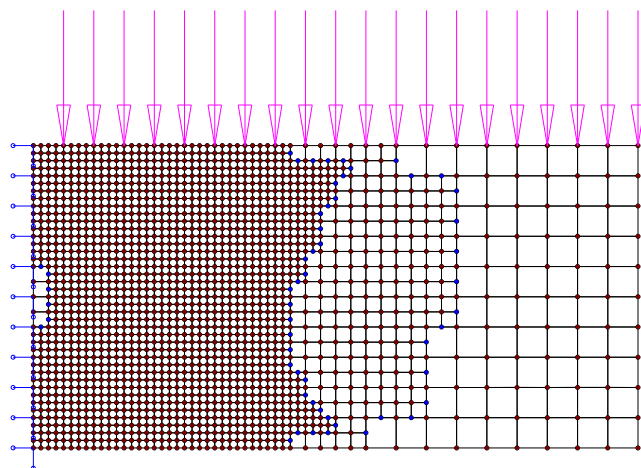
takie nazywane będą dalej węzłami przylegającymi. Metoda elementów skończonych nie gwarantuje w takiej sytuacji ciągłości pola przemieszczeń. Aby zgodność przemieszczeń w takim przypadku była możliwa do zapewnienia funkcje kształtu elementów skończonych muszą być tego samego stopnia. Ponadto dla zapewnienia zgodności przemieszczeń w takiej sytuacji konieczne jest wprowadzenie dodatkowych związków na przemieszczenia, aby w procesie odkształcania węzły przylegające niezmiennie znajdowały się na krawędzi. W naszym przykładzie węzeł 5 powinien leżeć niezmiennie na krawędzi 2-3 elementu 1. Żeby taką zgodność uzyskać niewiadome węzła leżącego na krawędzi nie mogą być już niezależnymi stopniami swobody, ale funkcjami przemieszczeń elementu na którego krawędzi węzeł leży. Tak więc wielkości elementowe, takie, jak np. macierz sztywności czy wektor elementowego obciążenia, będą musiały być przetransformowane do nowej przestrzeni przy uwzględnieniu związków przemieszczeń. Wymiar macierzy sztywności elementu z węzłami



Rysunek 6.15: Zagęszczanie izotropowe

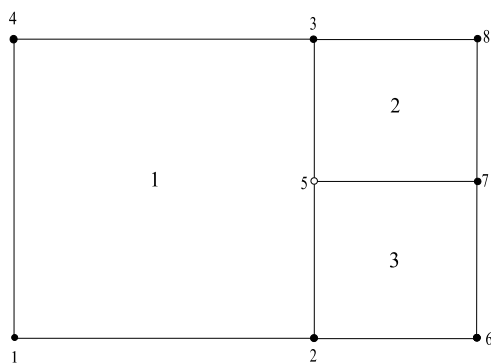


Rysunek 6.16: Wspornik pokryty regularną siatką elementów skończonych z liniową aproksymacją Lagrange4



Rysunek 6.17: Wspornik po wykonaniu zagęszczenia izotropowego Lagrange4

przylegającymi powiększa się o pewną liczbę stopni swobody elementu, do którego element dany przylega, pomijając jednocześnie stopnie samego węzła przylegającego, jak już wspomniano, nie będące wielkościami niezależnymi. W przypadku liniowych funkcji kształtu, z przykładu zamieszczonego na Rys. 6.18, łatwo zauważyć, że przemieszczenia węzła przylegającego 5 zależne muszą być tylko od węzłów 2,3, leżących na tej samej krawędzi, co węzeł 5. W praktyce podobnie będzie w przypadku innych, zwykle używanych wielomianowych funkcji kształtu. Przemieszczenia węzła przylegającego zależne będą od przemieszczeń węzłów elementu, definiujących tę krawędź do której węzeł przylega. W zaprezentowanym tutaj trywialnym przypadku widać, że aby była zapewniona zgodność przemieszczeń



Rysunek 6.18: Fragment siatki po wykonaniu zagęszczenia izotropowego. Węzeł 5 jest węzłem przylegającym Lagrange4

węzła 5, muszą one być średnią arytmetyczną przemieszczeń węzłów 2 i 3. Dla problemu przepływu ciepła wektor stopni swobody elementu 3 dla aproksymacji związanej ma zatem postać $\hat{\mathbf{t}}_3^e = \{t_2, t_6, t_7, t_5, t_3\}$. Nieprzetworzona macierz przewodnictwa cieplnego tego elementu określona jest wzorem

$$\mathbf{H}_3^e = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & 0 \\ h_{21} & h_{22} & h_{23} & h_{24} & 0 \\ h_{31} & h_{32} & h_{33} & h_{34} & 0 \\ h_{41} & h_{42} & h_{43} & h_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.11)$$

Macierz transformacji wielkości z aproksymacji związanej do pierwotnych wielkości ma postać:

$$\mathbf{E}_{3,5}^e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.12)$$

Indeksy oznaczają transformacje dla elementu 3 i węzła przylegającego 5. Transformacja wielkości z transformacji związanych do zwykłych realizowana jest następująco

$$\mathbf{t}_3^e = \begin{bmatrix} t_2 \\ t_6 \\ t_7 \\ \frac{t_2+t_3}{2} \\ 0 \end{bmatrix} = \mathbf{E} \begin{bmatrix} t_2 \\ t_6 \\ t_7 \\ t_5 \\ t_3 \end{bmatrix} \quad (6.13)$$

Transformacja macierzy pojemności wygląda następująco

$$\mathbf{H}^e = \mathbf{E} \cdot \hat{\mathbf{H}} \cdot \mathbf{E}^T \quad (6.14)$$

W przypadku gdy element i ty posiadałby p węzłów przylegających, wówczas macierz transformacji należy wyznaczyć następująco:

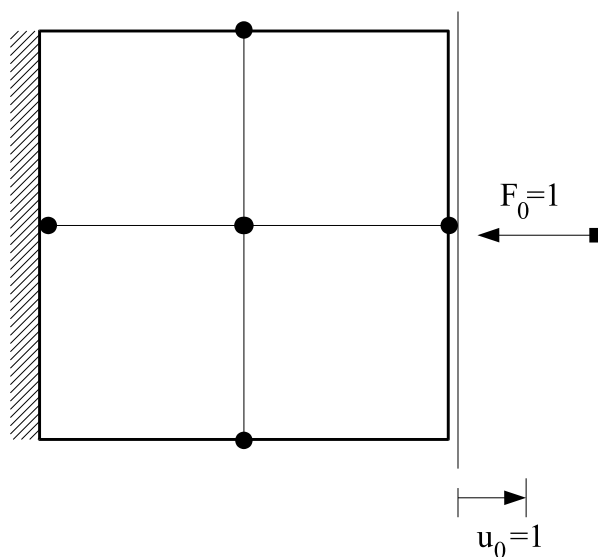
$$\mathbf{E}_i^e = \mathbf{E}_{i,1}^e \mathbf{E}_{i,2}^e \dots \mathbf{E}_{i,p}^e \quad (6.15)$$

6.9 Przykłady numeryczne

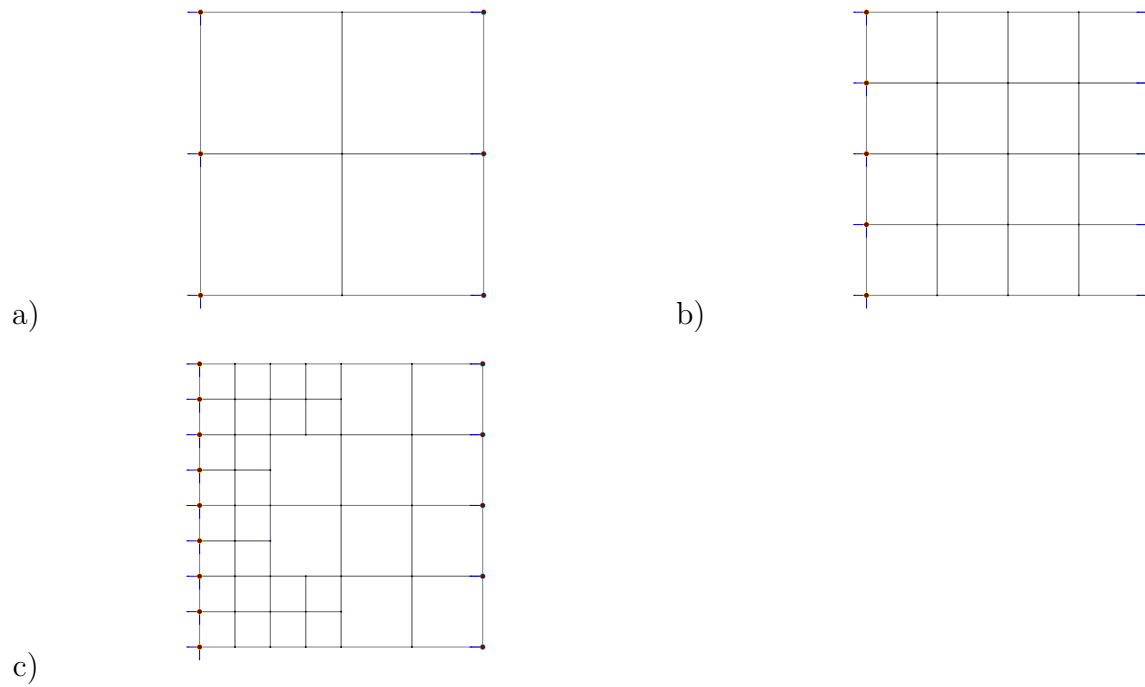
Wykonano szereg przykładów numerycznych z wykorzystaniem technik adaptacyjnych *h-refinement*. Zastosowano dwie strategie adaptacyjne. Pierwsza stosowana dla siatek regularnych polegająca na podziale prostokątnych elementów na cztery mniejsze. Druga metoda stosowana dla siatki trójkątnej, nieregularnej polega na wygenerowaniu nowej siatki. Poprawność implementacji sprawdzono rozwiązując przykłady zamieszczone w pracy [50]. Testując przykład ze strony 411, Fig 15.7 otrzymano takie same wartości relatywnej normy energii η w każdej iteracji.

6.9.1 Przykład numeryczny - test relaksacji

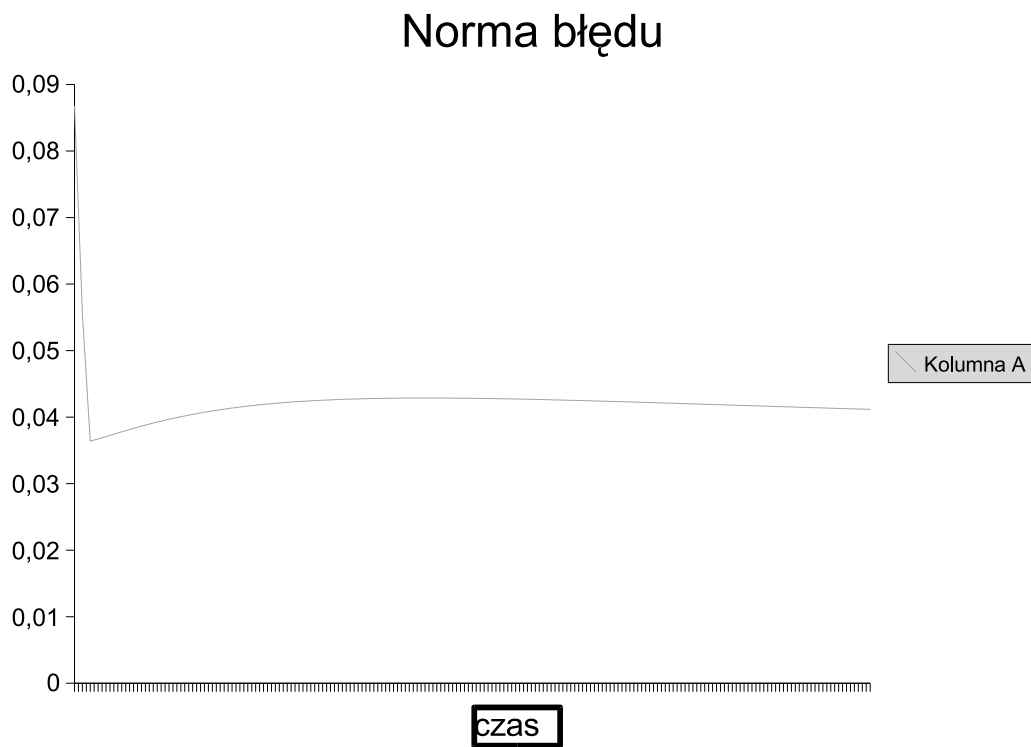
Analizowano w tym przykładzie wspornik zamieszczony na Rys. 6.19. Jako obciążenie geometryczne przyjęto przemieszczenie prawej krawędzi wspornika $u_0 = 1$ stały w w całym przedziale czasu. Na tej samej krawędzi działa na konstrukcję strumień ciepła $F_0 = 1$. Pozostałe dane zadania: materiał termolepkosprężysty, $\tau_0 = 30s$. 100 równych kroków czasowych zastosowano w tym teście. Użyto następujących wartości stałych materiałowych $E^0 = 500$, $E^\infty = 100$, $\rho = 1$, $\nu = 0.3$, $\kappa = \frac{\lambda}{\rho c} = 1$. Na utwierdzonym brzegu zadano termiczne warunki brzegowe $T = 0$. Dyskretyzację MES stanowią cztery elementy Lagrange'a z liniowymi funkcjami kształtu (czterowęzłowe). W pierwszym kroku czasowym analizy termolepkosprężystej okazało się, że relatywna norma błędu nie spełnia zakładanej wartości dopuszczalnej ($\bar{\eta} = 5\%$). Skutkowało to uruchomienie procedury adaptacyjnej. Poszczególne etapy adaptacji uwidacznia Rys. 6.20. Za trzecim razem jakość siatki spełniła już założoną normę w pierwszym kroku czasowym. W następnych krokach czasowych również norma błędu nie przekraczała zakładanej wartości granicznej i obliczenia mogły być przy zastosowaniu tej samej siatki. Zastosowano tu metodę restartu czyli rozpoczynano analizę od początku po zmodyfikowaniu siatki. Zachowanie się normy błędu w procesie relaksacji pokazano na Rys. 6.21



Rysunek 6.19: Schemat statyczny wspornika

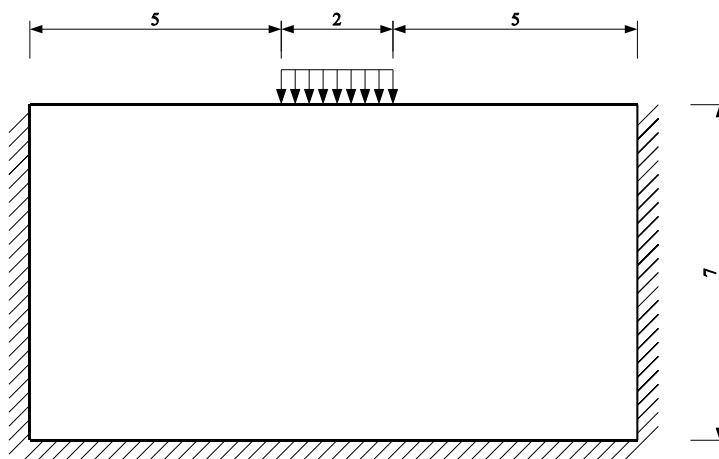


Rysunek 6.20: Kolejne fazy procesu poprawianie siatki. Wartości relatywnej normy energii dla poszczególnych siatek: a) $\eta = 8.7\%$, b) $\eta = 5.6\%$, c) $\eta = 3.7\%$



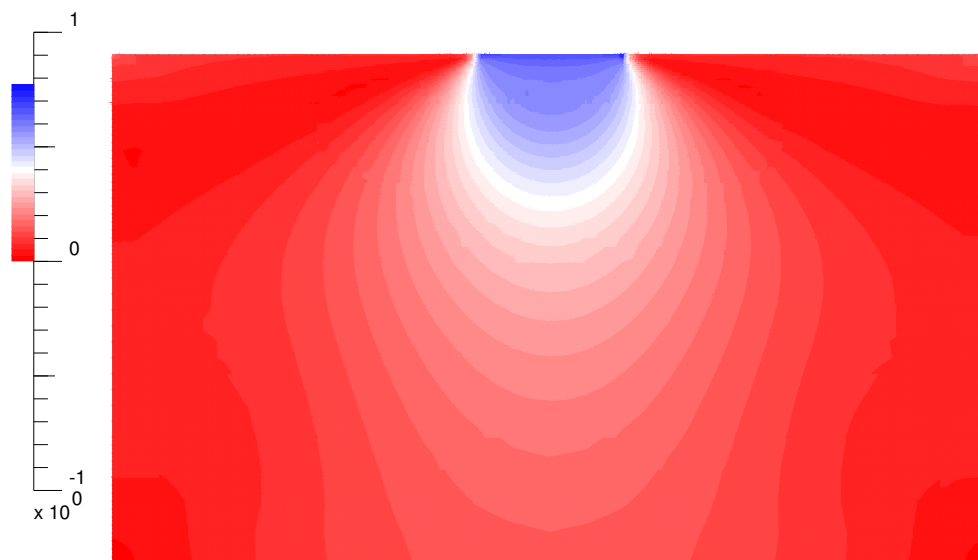
Rysunek 6.21: Relaksacja relatywnej normy błędu

6.9.2 Przykład numeryczny - Połprzestrzeń sprężysta

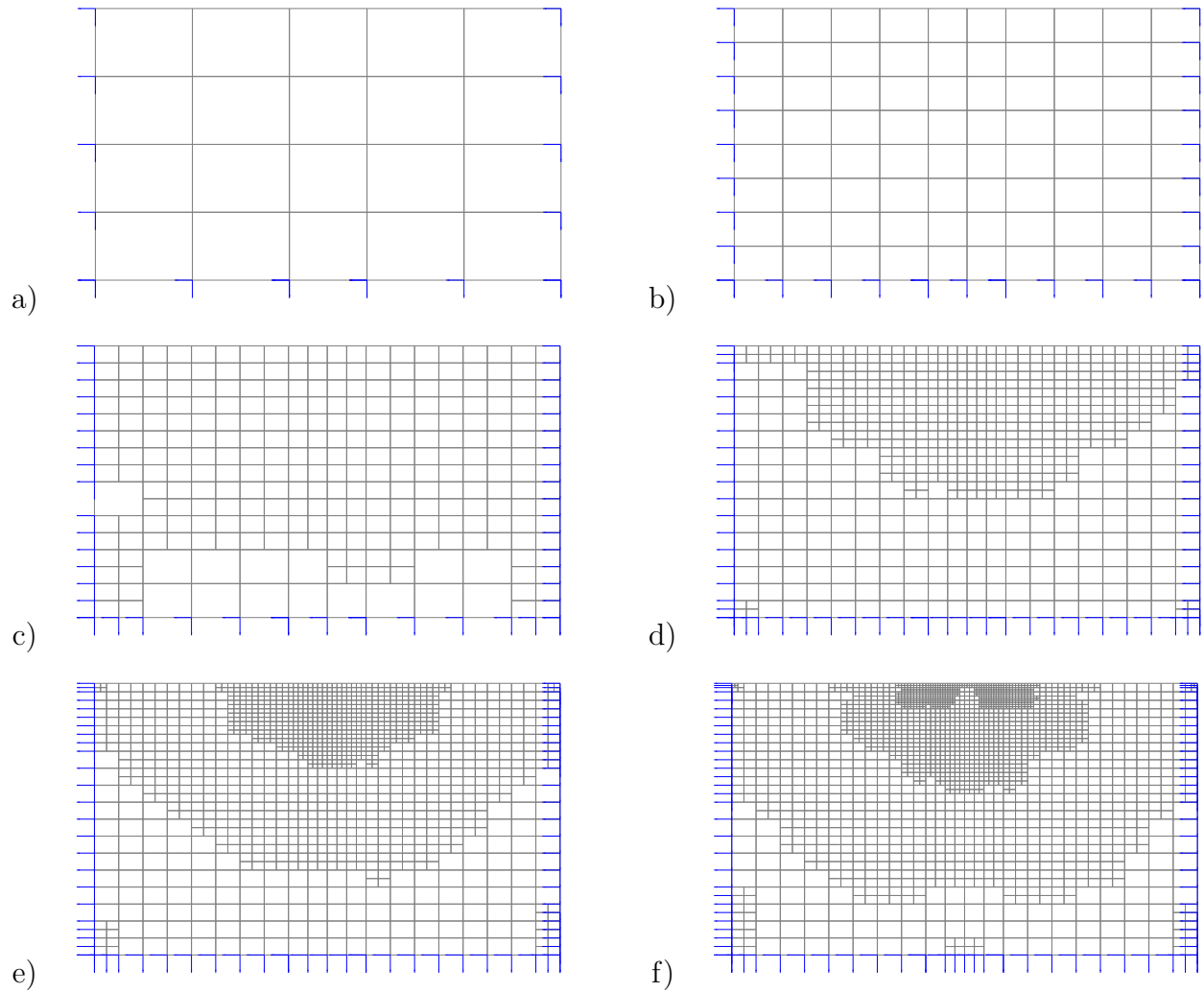


Rysunek 6.22: Schemat zadania

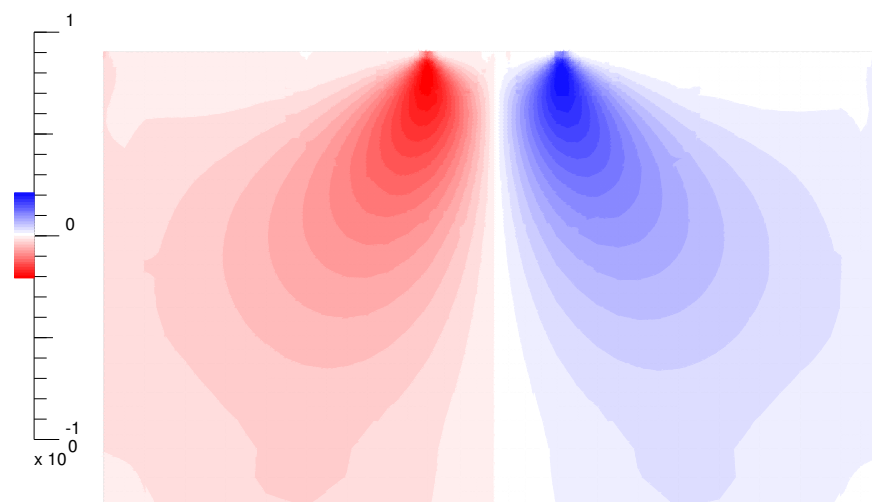
Podział elementów prostokątnych na mniejsze jest strategią adaptacji siatki zastosowana w tym przykładzie. Schemat zadania przedstawiony został na rysunku 6.22. Przyjęto materiał liniowo sprężysty o właściwościach $E = 210000$, $\nu = 0.3$. Redukcję relatywnej normy błędu poniżej zakładanego progu 5% osiągnięto po pięciu iteracjach. Siatki, oraz wartości normy błędu przedstawione są na rysunku 6.24. System VFEM++ posiada mechanizmy graficznej prezentacji wyników. Możliwe jest tworzenie map warstwicznych wielkości wynikowych. Dla tego przykładu załączono mapy warstwiczne naprężeń zredukowanych Hubera-Misesa 6.23, oraz naprężeń ścinających 6.25.



Rysunek 6.23: Mapa naprężeń zredukowane Mizesa

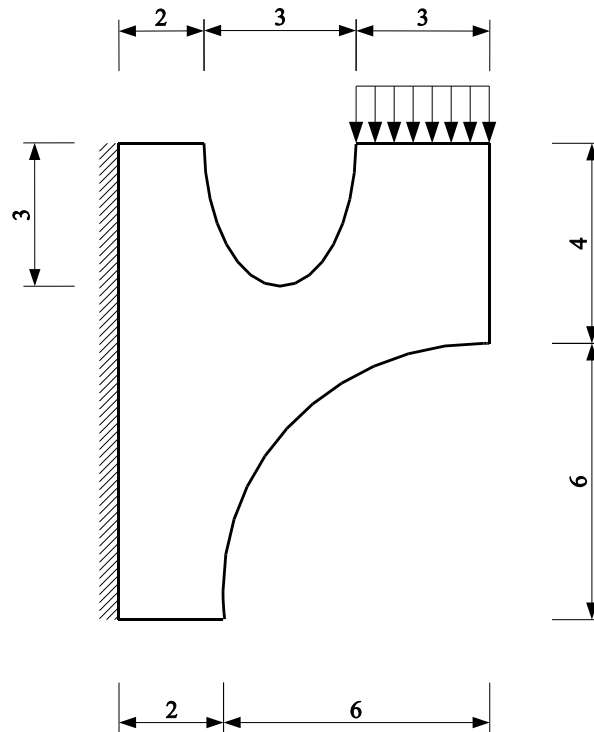


Rysunek 6.24: Kolejne fazy procesu poprawianie siatki. Wartości relatywnej normy energii dla poszczególnych siatek: a) $\eta = 39\%$, b) $\eta = 24.1\%$, c) $\eta = 14.1\%$, d) $\eta = 8.4\%$, e) $\eta = 5.2\%$, f) $\eta = 3.7\%$



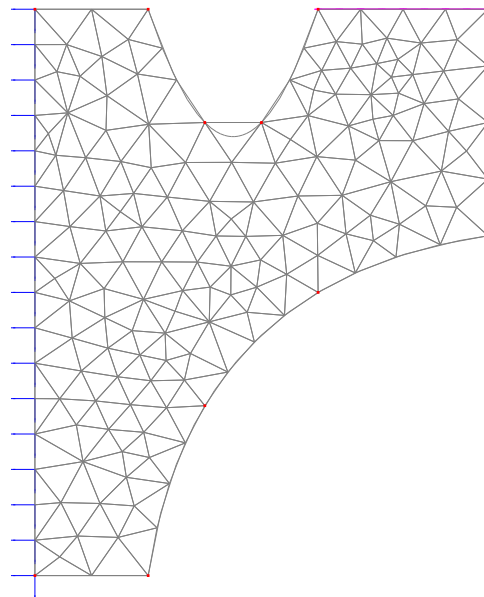
Rysunek 6.25: Mapa naprężeń ścinających

6.9.3 Zagęszczanie siatek przez ponowną generację siatki

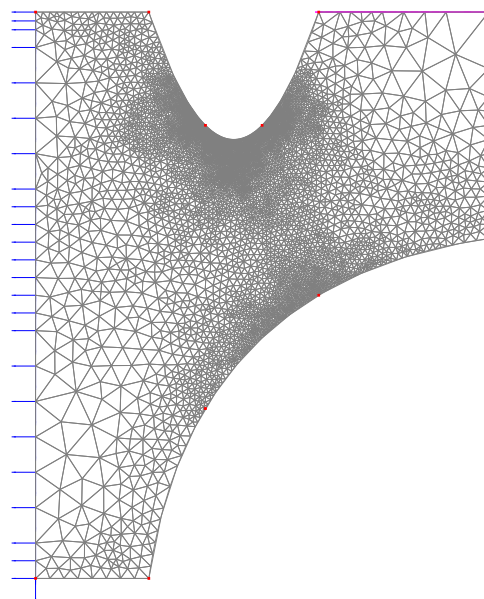


Rysunek 6.26: Schemat statyczny

Ponowna generacja siatki zmienną gęstością węzłów skorelowaną z rozkładem błędów jest strategią adaptacji siatki stosowaną na nieregularnych siatkach trójkątów. Strategia stosowana jest w zadaniach z nieregularną siatką trójkątną. Stworzony w ramach tej pracy system ma możliwość generowania siatek trójkątnych. Dodano również techniki pozwalające na poprawianie rozwiązania uzyskanego przy pomocy tych typów siatek. Schemat statyczny zadania przedstawiony został na Rys. 6.26. Zastosowano materiał sprężysty o następujących własnościach: $E = 210000$, $\nu = 0.3$. Obszar został pokryty siatką trójkątów o podobnych wymiarach przy użyciu triangulacji Delaunaya (Rys. 6.27a). Do obliczeń zastosowano element trójkątny CST. Następnie w oparciu o uzyskany ze wzoru (6.9) rozkład przewidywanych wielkości elementów. Wygenerowano nową siatkę (Rys. 6.27b). Wartość relatywnej normy energii dla tej siatki $\eta = 3.4\% < 5\%$ pozwolił zakończyć proces już po pierwszej iteracji. Dla przykładu podano kilka map warstwowych naprężeń Rys. 6.28, 6.29.

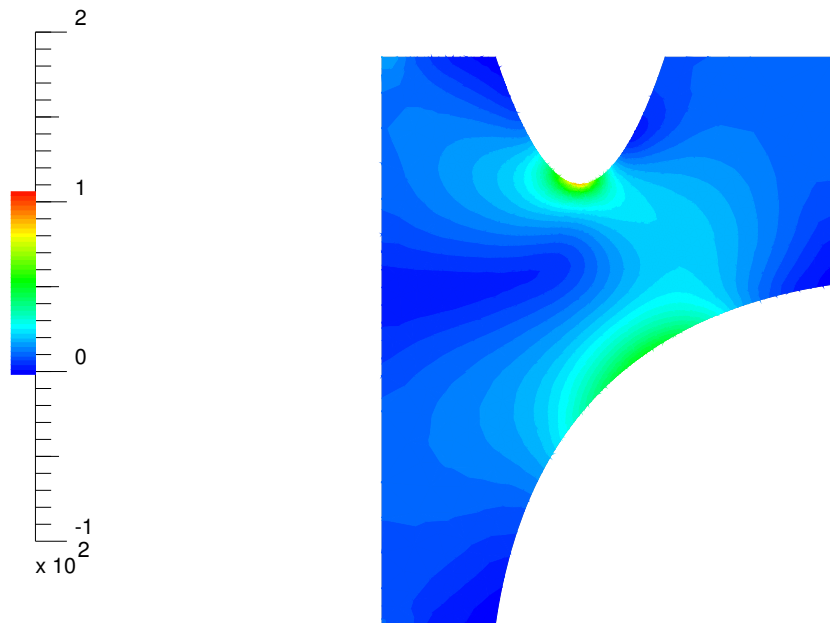


a)

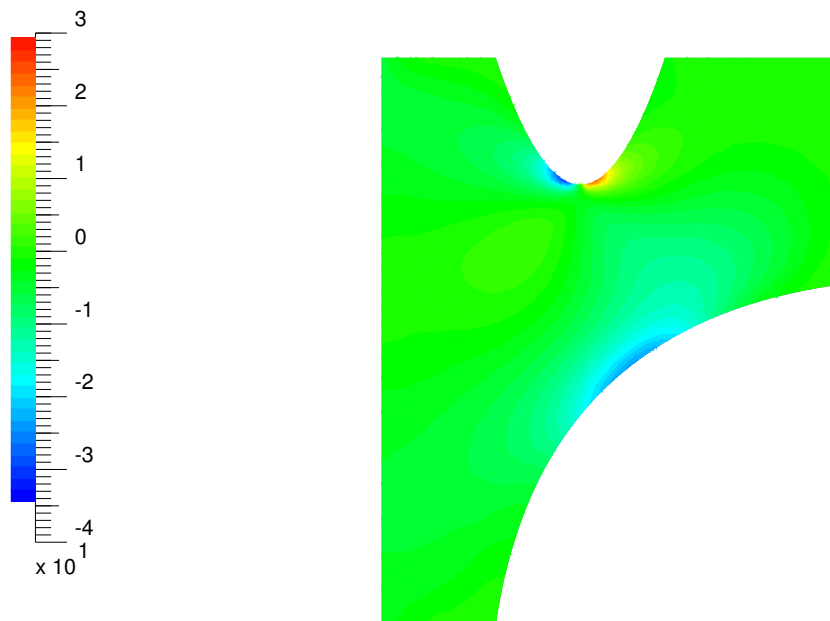


b)

Rysunek 6.27: Kolejne fazy procesu poprawianie siatki. Wartości relatywnej normy energii dla poszczególnych siatek: a) $\eta = 27.1\%$, b) $\eta = 3.4\%$



Rysunek 6.28: Naprężenia zredukowane w/g hipotezy Hubera-Misesa



Rysunek 6.29: Naprężenia ścinające

Proces poprawiania jakości siatki trójkątów przez ponowną generację jest bardziej zbieżny, niż poprzednio opisywany proces podziału siatek regularnych. Jeśli konstrukcja nie jest obciążona obciążeniem skupionym, wówczas zwykle wystarcza jedna lub dwie iteracje, podczas gdy w analogicznym zadaniu z siatką regularną potrzebnych może być dwa, trzy razy więcej iteracji. Procedura generowania nowej siatki trójkątnej, w oparciu o rozkład błędu brany ze starej siatki, jest jednak czasochłonna i trudno jednoznacznie wskazać większą efektywność którejś z metod. Poważne problemy sprawia również obciążenie skupione, które sprawia, że proces poprawiania siatki jest wolno zbieżny i prowadzi to powstania zadań o przesadnie dużej liczbie stopni swobody. Obie techniki w efekcie prowadzą do otrzymania zadań o zbliżonej liczbie stopni swobody. Koniecznością jest stosowanie procedur optymalizacji profilu macierzy. Numeracja węzłów powstającej w procesie adaptacji nowej siatki, jest przypadkowa. Prowadzi to do powstania zbyt dużych macierzy a w praktyce nawet do niemożności rozwiązania zadania, z powodu zbyt długiego czasu dekompozycji. W tworzonym systemie używana jest klasyczna metoda Reversal Cuthill-McKee (RCM), [83].

Wnioski

Celem pracy było zastosowanie podejścia obiektowego do stworzenia systemu MES służącego do analizy wielu problemów mechaniki konstrukcji, a w szczególności do analizy układów termolepkosprężystych, z uwzględnieniem zniszczenia. Do systemu włączono również metody optymalizacji oraz szacowania niezawodności konstrukcji. System napisany został w języku C++ przy pomocy środowiska programistycznego Visual C++ 6.0 professional edition. Systemem VFEM++ można analizować następujące rodzaje konstrukcji: belki, ramy i kratownice płaskie oraz przestrzenne, tarcze w płaskich stanach naprężenia oraz odkształcenia, płyty średniej grubości bazujące na założeniach kinematycznych Reissnera-Mindlina oraz powłoki obrotowo symetryczną i średniej grubości typu Ahmada. Konstrukcje można poddać następującym rodzajom analiz: statyka liniowa, dynamika liniowa, statyka liniowa ze zniszczeniem, statyka nieliniowa z nieliniowością materiałową (płaskie stany) oraz geometrycznie nieliniową (kratownice), termolepkosprężystość ze zniszczeniem oraz niestacjonarny przepływ ciepła. Zaimplementowano również możliwość liczenia gradientów funkcji wynikowych. W analizie wrażliwości stosowana była metoda DDM. Można stosować trzy rodzaje zmiennych projektowych: stałe materiałowe, parametry kształtu oraz parametry obciążenia. W systemie VFEM++ istnieją następujące rozkłady prawdopodobieństw zmiennych losowych: rozkład normalny, lognormalny, jednostajny, Gumbela, Frecheta oraz Weibula. Do wyznaczenia rozkładu błędów konstrukcji zastosowano estymator błędu Zienkiewicza-Zhu, na którym bazują algorytmy poprawiające jakość siatki. Stosuje się dwie metody poprawiania siatki : przez podział - dla siatek regularnych i przez ponowną generację dla siatek triangularyzowanych.

Elementami oryginalnymi pracy według autora są:

- Opracowanie koncepcji środowiska obliczeniowego MES z zastosowaniem podejścia obiektowego.
- Efektywna realizacja tej koncepcji w środowisku programistycznym języka C++.
- Wyprowadzenie równań MES dla pewnego modelu termolepkosprężystego.

- Wyprowadzenie równań MES dla wrażliwości parametrycznej oraz wrażliwości na parametry kształtu.
- Implementacja numeryczna zagadnienia termolepkosprężystego modelu materiału w obiektowym systemie MES.
- Opracowanie struktur obiektowych dla analizy optymalizacyjnej i niezawodnościowej.
- Włączenie analiz optymalizacyjnej i niezawodnościowej do stworzonego środowiska obliczeniowego.
- Policzenie szeregu przykładów numerycznych z zakresu analizy optymalizacyjno-niezawodnościowej konstrukcji termolepkosprężystych ze zniszczeniem oraz krytyczna ocena otrzymanych rezultatów.

Przeprowadzone eksperymenty numeryczne, doświadczenia nabyte podczas budowy oprogramowania oraz rozważania teoretyczne pozwalają sformułować następujące wnioski:

1. Podejście obiektowe, zastosowane do budowy systemu MES pozwala na bardzo efektywne, szybkie stworzenie złożonego systemu analizy MES.
2. Struktury danych obiektowego systemu MES odzwierciedlają naturalne związki pomiędzy pojęciami dziedziny problemu, co prowadzi do otrzymania znacznie bardziej przejrzystego kodu i lepszej diagnostyki, niż w tradycyjnym podejściu strukturalnym.
3. Stosowanie funkcji wirtualnych, o większej niż kilka linii objętości kodu, które nie mogłyby być rozwijane w miejscu (np. metoda obliczająca macierz sztywności elementu) nie powoduje zauważalnego zmniejszenia efektywności programu w stosunku do metod niewirtualnych.
4. Dla zrealizowania mechanizmu polimorficzności mniejszych składowych funkcji, zawierających w swoim ciele tylko kilka linii (np. składowa obiektu aproksymacja obliczająca wartości funkcji kształtu), należy zamiast funkcji wirtualnych rozważyć zastosowanie szablonów. Rozwiązanie takie pozwala kompilatorowi na umieszczenie kodu funkcji w miejscu wywołania, zamiast wywołania funkcji, co może przyczynić się do poprawy efektywności programu.
5. W przeprowadzonych eksperymentach numerycznych uzyskano podobną efektywność numeryczną, jak w analogicznym programie, napisanym w języku FORTRAN 95.
6. Podczas tworzenia systemu VFEM++ zwracano baczna uwagę na stosowanie rozwiązań zawartych w standardzie języka C++, dzięki czemu kod systemu jest przenośny i przeniesienie go do innego systemu operacyjnego zajmuje niewiele czasu.
7. Integracja analizy skończenie elementowej, optymalizacji i analizy niezawodnościowej prowadzi do zwiększenia efektywności obliczeń, dzięki unikaniu konieczności komunikacji poszczególnych modułów, poprzez wymianę plików pomiędzy modułami MES i niezawodności czy optymalizacji. Jest to szczególnie istotne w przypadku metody Monte Carlo gdzie konieczność wykonania np. miliona analiz MES sprawia, że każda, nawet drobna optymalizacja kodu, może przekładać się na godziny obliczeń.

8. Nielokalne sformułowanie modelu zniszczenia znacznie poprawia obiektywność obliczeń sprawiając, że rozwiązanie jest mniej czułe na rodzaje przyjętej siatki MES.
9. Asymptotyczne wartości przemieszczeń w procesie pełzania i naprężeń w teście relaksacji, dla czasu dążącego do nieskończoności nie zależą od kształtu programu obciążeń w funkcji czasu, tylko od granicy wartości obciążeń dla czasu dążącego do nieskończoności.
10. Wyprowadzony w ramach pracy termolepkosprężysty model materiału ma tę cechę, że temperatura ma wpływ na własności mechaniczne, ale nie na odwrót. Układ równań takiego problemu może być rozwiązany niezależnie tj. najpierw problem niestacjonarnego przepływu ciepła, a następnie zagadnienie lepkosprężyste, w oparciu o wyznaczoną temperaturę.
11. Warto zaimplementować dwie wersje równań MES dla problemu termolepkosprężystego. Pierwszą, z macierzą konsystentną, gdzie niewiadomy wektor przyrostów przemieszczeń jest tylko po lewej stronie. Niepotrzebna jest wówczas iteracja na kroku czasowym do znalezienia niewiadomej, ale za to konieczna jest dekompozycja macierzy konsystentnej. W drugiej wersji równania, niewiadomy wektor jest również po prawej stronie równania, konieczna jest iteracja do rozwiązania równania na kroku, ale za to nie trzeba dekomponować dużej macierzy. Przy dużych zadaniach może się okazać, że wersja z iteracją jest bardziej efektywna, niż wersja z macierzą konsystentną.
12. Istotnym problemem przy paralelizacji obliczeń jest nie tylko algorytm, ale także dostęp do wspólnych zasobów. W łatwym z pozoru do paralelizacji algorytmie agregacji macierzy sztywności uzyskano na maszynie dwuprocessorowej przyspieszenie 1.7 w stosunku do jednoprocessorowej, co było spowodowane koniecznością kolejkowania dostępu wątków do macierzy sztywności, w celu akumulacji elementowej macierzy sztywności.
13. Przy stosowaniu technik polepszania jakości siatki, koniecznością jest stosowanie procedur optymalizacji profilu macierzy. Numeracja węzłów, powstającej w procesie adaptacji nowej siatki, jest przypadkowa. Prowadzi to do powstania zbyt dużych macierzy, a w praktyce nawet do niemożności rozwiązania zadania, z powodu zbyt długiego czasu dekompozycji macierzy.

Bibliografia

- [1] **A. Einstein**, *Dialectica*, 2:320, 1948.
- [2] **W.H. Frey** *Selective Refinement: A new strategy for automatic node placement in graded triangular meshes*, Int. J. Num. Meth. in Eng., vol 24, 2183-2200, 1987.
- [3] **M. Flower** *Comparision of Object-Oriented Analysis and Design Methods*, A white paper from IntelliCorp, 1993.
- [4] **B. Stroustrup**, *Jezyk C++*, Wydawnictwa Naukowo-Techniczne 2002.
- [5] **P. Coad, Edward Yourdon** *Analiza obiektowa*, Oficyna Wydawnicza READ ME, 1996.
- [6] **P. Coad, Edward Yourdon** *Projektowanie obiektowe*, Oficyna Wydawnicza READ ME, 1996.
- [7] **P. Coad, J. Nicola** *Programowanie obiektowe*, Oficyna Wydawnicza READ ME, 1996.
- [8] **J. Martin, J. J. Odell** *Podstawy metod obiektowych*, Wydawnictwa Naukowo Techniczne, 1997.
- [9] **S. B. Lippman**, *Model obiektu w C++*, Wydawnictwa Naukowo-Techniczne 1999.
- [10] **J. Liberty**, *C++ księga eksperta*, Helion 1999.
- [11] **B. W. R. Forde, R. O. y Foschi, S. F. Stierner**, *Object-Oriented Finite Element Analysis*. Computers and Structures, 34, 355-374, 1990.
- [12] **G.L. Fenves**, *Object-oriented programming for engineering software development*, Engineering with computers, 6,1-15, 1990.
- [13] **G. R. Miller**, *A LISP based object oriented approach to structural analysis* Engineering with Computers, 4, 197-203, 1988.
- [14] **G. R. Miller**, *An Object-Oriented Approach to Structural Analysis and Design* Computers and Structures, 40, 75-82, 1991.
- [15] **T. Zimmermann, Y. Dubois-Pèlerin, P. Bomme**, *Object-Oriented finite element programming: I Governing principles*, Computer Methods in Applied Mechanics and Engineering, 98, 291-303, 1992.

- [16] **Y. Dubois-Pèlerin, T. Zimmermann**, *Object-Oriented finite element programming: III An efficient implementation in C++*, Computer Methods in Applied Mechanics and Engineering. 108, 165-183, 1993.
- [17] **X. A. Kong, D. P. Chen**, *An Object-Oriented Design of FEM Programs*, Computers and Structures. 57, 157-166, 1995.
- [18] **R.R Gajewski**, *An object-oriented approach to finite element programming*, Civil-Comp, 107-113, 1996.
- [19] **R.R Gajewski, P. Lompies**, *Object-oriented implementation of bandwidth, profile and wavefront reduction algorithms*, Civil-Comp, 115-119, 1996.
- [20] **R.R Gajewski, T. Kowalczyk**, *A prototype object-oriented finite element method program: Class hierarchy and graphic user interface*, CAMES, 3, 65-74, 1996.
- [21] **T. Zimmermann, D. Eyheramendy**, *Object-Oriented finite element. I Principles of symbolic derivations and automatic programming*, Computer Methods in Applied Mechanics and Engineering. 132, 259-276, 1996.
- [22] **D. Eyheramendy, T. Zimmermann**, *Object-Oriented finite element. II A symbolic environment fo automatic programming*, Computer Methods in Applied Mechanics and Engineering. 132, 277-304, 1996.
- [23] **Y. Dubois-Pèlerin, P. Pegon**, *Improving modularity in object-oriented finite element programming*, Communications in Numerical Methods in Engineering. Vol. 13, 193- 198. 1997.
- [24] **R.I. Mackie**, *Using objects to handle complexity in Finite Element software*, Engineering with Computers. 13, 99-111, 1997.
- [25] **Y. Dubois-Pèlerin M. Topcu** *Object-oriented programming in nonlinear finite element analysis*, Computer & Structures, 67:225-242, 1998.
- [26] **Th. Zimmermann P. Bomme, D. Eyheramendy, L. Vernier S. Commend** *Aspects of an object-oriented finite element environment*, Computer & Structures, 68:1-16, 1998.
- [27] **D. Abury, B. Tie**, *Object-oriented programming in advanced structural computations*, ECOMMAS 2000, Barcelona, 11-14 September 2000.
- [28] **L. Gil, G. Bugada**, *An object-oriented programming strategy for the implementation in C++ of the sensitivity analysis using a structural non-linear material model*, ECOMMAS 2000, Barcelona, 11-14 September 2000.
- [29] **R.I. Mackie** *An object-oriented approach to calculation control in finite element programs*, Computer & Structures, 77:461-474, 2000.
- [30] **R. I. Mackie** *Finite Element Analysis*, Saxe-Coburg Publications, 2001.
- [31] **L. Yu A.V. Kumar** *An object-oriented modular framework for implementing the finite element method*, Computer & Structures, 79:919-428, 2001.
- [32] **B. Patzák, Z. Bittnar** *Design of object oriented finite element code*, Advances in Engineering Software, 32, 759-767, 2001.
- [33] **F.J. Lingen** *Design of an object oriented Finite Element package for parallel computers*, The Netherlands 2000.

- [34] **C. Farhat F.X Roux** *Implicit Paralell processing in structural mechanics*, Computational Mechanisc Advances, 2:1-124, 1994
- [35] **H. Poon, M. Ahmad** *A material point time integration procedure for anisotropic, thermo rheologically simple, viscoelastic solids.*
- [36] **Zocher MA, Groves SE, Hilton H** *A three-dimensional finite element formulation for thermoviscoelastic orthotropic media*, J. Num Meth., Eng 40:2267-2288, 1997.
- [37] **M. Kleiber et al** *Parameter sensitivity in nonlinear mechanics*, Wiley 1997.
- [38] **M. Kleiber, T.D. Hien** *The Stochastic Finite Element Method*, Wiley 1992.
- [39] **M. Kleiber et al** *Mechanika Techniczna, tom 11 - Komputerowe Metody Mechaniki Ciał Stałych*, PWN Warszawa 1995.
- [40] **M. Kleiber, C. Woźniak** *Nonlinear Mechanics of Structures*, Kluwer-PWN 1991.
- [41] **K. Schittkowski** *Nlpgl: A fortran subroutine solving constrained nonlinear programming problems*, Annals of Operations Research 5:485-500, 1986.
- [42] **J. Argyris, J. St. Doltinis, V. D. Da Silvia** *Constitutive modelling and computation of non-linear viscoelastic solids*, Comp. Meth. In App. Mech 88, 1991.
- [43] **JC Simo, RL Taylor** *Consistent tangent operator for rate independent elastoplasticity*, Comp. Meth. In App. Mech 48:182-118, 1985.
- [44] **M. Henriksen** *Non-linear viscoelastic stress analysis - a finite element approach.*, Comp & Struct., 18:133-139.
- [45] **H.S.Carslaw, J.C. Jaeger** *Conduction of heat in solids*, Clarendon Press, Oxford 1959.
- [46] **W. Hackbush** *Iterative Solution of Large Sparse Systems of Equations*, Springer Verlag 1994.
- [47] **B.H.V. Topping J. Muylle P. Iványi R. Putanowicz B. Cheng** *Finite Element Mesh Generation*, Saxe Coburg Publications 2004.
- [48] **W. Sosnowski** *Numeryczna symulacja, analiza wrażliwości i optymalizacja nieliniowych procesów deformacji konstrukcji*, Wydawnictwo Akademii Bydgoskiej 2003.
- [49] **Z. Mróz, M.P. Kamat, H. Plaut** *Sensitivity analysis and optimal design of non-linear beams and plates*, J. Struct. Mech., 13:245-266 1985.
- [50] **O. C. Zienkiewicz R.L. Taylor** *The Finite Element Method*, Butterworth-Heinemann, 2000.
- [51] **M.A. Crisfield** *Non-linear Finite Element Analysis od Solids and Structures*, John Wiley & Sons, 1991.
- [52] **R.T Haftka, Z. Gurdal** *Elements of Structural Optimization*, Kluwer Academic Publishers, 1992.
- [53] **J.S. Arora** *Introduction to optimum design*, McGraw Hill Book Company, 1988.
- [54] **R. Stocki** *Optymalizacja niezawodnościowa konstrukcji prętowych w zakresie dużych przemieszczeń. Teoria i program komputerowy.*, Prace IPPT nr 13, 1999.
- [55] **T. Abdo, R. Rackwitz** *Reliability of Uncertain Structural Systems*, Finite Elements in Eng. Appl., 161-176, 1990.

- [56] **Ž. Bažant L. Cedolin** *Stability of Structures*, Oxford University Press, 1991.
- [57] **L.M. Katchanov** *On Creep rupture time*, Izv. Acad. Nauk. SSSR, Otd. Techn Nauk. No. 8, 1958, 26-31 1958.
- [58] **L.M. Katchanov** *The Theory of Creep*, Nauka, Moscow, 1960, (In Russian, English translation by Kennedy (Ed.), National Lending Library, Boston Spa, England, 1967). 1958, 26-31 1958.
- [59] **L.M. Katchanov** *Introduction to Continuum Damage Mechanics*, Kluwer Academic Publishers 1986.
- [60] **J. Lemaitre L. Chaboche** *Aspect phenomenologique de la rupture per endommagement*, J. Mécanique Appliquée, 2:317-365, 3, 1978.
- [61] **J. Lemaitre L. Chaboche** *Mechanics of solid materials*, Cambridge University Press 1990.
- [62] **L. Chaboche** *Continuum Damage Mechanics*, I, II J. Appl. Mech, 65-72, 1988.
- [63] **Y.N. Rabotnov** *Creep Problems in Structural Members*, North-Holland, Amsterdam, 1969.
- [64] **Y.N. Rabotnov**. *On the equations of state for creep*. In P. in Applied Mechanics, editor, Prager Anniversary Volume, pages 307-. mcm, 1963.
- [65] **F.K.G. Odquist** *Mathematical Theory of Creep and Creep Rupture*, Clarendon Press, Oxford, 1974.
- [66] **A. Kirsch** *Creep tests*, Advances in Creep Design, Appl. Science Publishers, London, 1971.
- [67] **F.A. Leckie, D.R Hayhurst** *Constitutive Equations for Creep Rupture*, Acta Metallurgica, 25:1059-1070, 1977.
- [68] **F.A. Leckie, D.R Hayhurst** *Creep Rupture in Structures*, Proc. R. Lond., A340, 232-347, 1974.
- [69] **Y. Weitsman** *A Continuum Damage Model for Viscoelastic Materials*, J. Appl. Mech, 733-780 1988.
- [70] **M. Basista W.K. Nowacki(eds.)** *Modeling of damage and fracture processes in engineering materials*, Trends in Mechanics of Materials IPPT PAN, 1999.
- [71] **W. Derski, S. Ziemba** *Analiza Modeli Reologicznych*, PWN, 1968. 1999.
- [72] **G. van Zijl** *Computational Modeling of Masonry Creep and Shrinkage*, Meinema BV, Delft, The Netherlands, 1999.
- [73] **P. Ladeveze, D. Leguillon** *Error estimate procedure in the finite element method and application*, Journal of numerical Analysis, 20:485-509, 1983.
- [74] **L. Gallimard, P. Ladeveze and J. P. Pelle** *Error estimation and adaptivity in elastoplasticity*, International Journal for Numerical methods in Engineering, 39:189-217, 1996.
- [75] **C. Comi, U. Perego** *Error estimation and adaptivity in elastoplasticity*, International Journal for Numerical methods in Engineering, 39:189-217, 1996.
- [76] **M. Jirásek and T. Zimmermann** *Rotating crack model with transition to scalar damage: I. Local formulation, II. Nonlocal formulation and adaptivity*, LSC Internal Report 97/01, 1997.

- [77] **A. Huerta, A. Rodríguez-Ferran, P. Díez, J. Sarrate** *Adaptive finite element strategies based on error assessment*, International Journal for Numerical Methods in Engineering 46, 1803-1818, 1999.
- [78] **A. Rodríguez-Ferran, A. Huerta** *Error Estimation and adaptivity for nonlocal damage models*, International Journal of Solids and Structures, 37, 7501-7528, 2000.
- [79] **M. di Prisco, L. Ferrarra, F. Meftah, J. Pamin, R. do Borst, J. Mazars, J. M. Reynouard** *Mixed mode fracture in plain and reinforced concrete: some results on benchmark tests*, International Journal of Fracture, 103, 127-148, 2000.
- [80] **H. Askes, L. J. Sluys, B.B.C. de Jong** *Remeshing techniques for r-adaptive and combined r/h adaptive analysis with application to 2D/3D crack propagation*, Structural Engineering and Mechanics, 12, 459-474, 2001.
- [81] **C. Comi, U. Perego** *Finite element strategies for damage assessment up to failure*, In Proceedings of the 6th National Congress SIMAI, Chia Laguna, Italy, May 2002.
- [82] **B. Patzák, M. Jirásek** *Adaptive Resolution of Localized Damage in Quasi-brittle Materials*, Journal of Engineering Mechanics, 130, 720-732, 2004.
- [83] **E. Cuthill** *Several Strategies for Reducing the Bandwidth of Matrices*, Papers of the Symposium on Sparse Matrices and their Applications, IBM Thomas J. Watson Research Center, New York, 1971.
- [84] **P. Tazowski, M. Kleiber** *Parameter sensitivity formulation for viscoelastic structures*, Comp. Assisted Mech. and Eng. Sciences, 6, 1999.
- [85] **P. Tazowski, M. Kleiber** *Sensitivity analysis for viscoelastic bodies in object oriented finite element environment*, Comp. Assisted Mech. and Eng. Sciences, 10, 2003.