Instytutu Mechaniki Budowli
Wydział Inżynierii Lądowej
Politechniki Krakowskiej

INŻYNIERSKA PRACA DYPLOMOWA

ODPOWIEDŹ PRZEKROJU ZESPOLONEGO NA ZGINANIE UKOŚNE Z SIŁĄ OSIOWĄ

Michał Ziobro

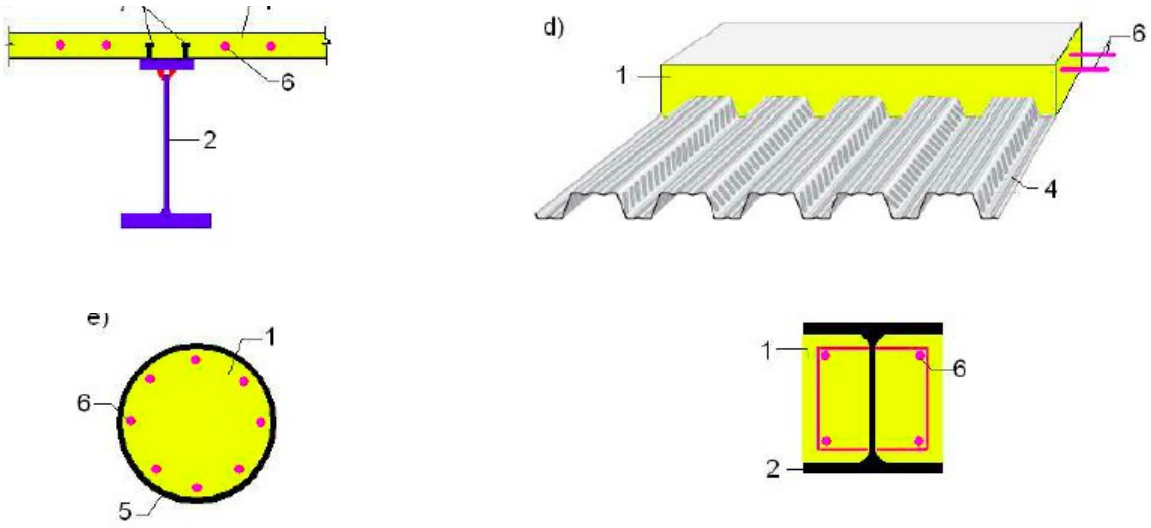
Promotor:
dr inż. Adam Zaborski

Kraków 2013

SPIS TREŚCI

1. WPROWADZENIE

Konstrukcje zespolone upowszechniły się w drugiej połowie XX wieku najpierw w Ameryce Północnej później w Europie między innymi za sprawą pojawienia się łączników ścinanych, które pozwoliły na osiągnięcie współpracy pomiędzy łączonymi materiałami. Poprzez konstrukcje zespolone rozumie się belki, płyty oraz słupy zespolone wykonane z co najmniej dwóch materiałów o różnych właściwościach fizycznych, przy czym powierzchnia każdego z tych materiałów musi być znacząca w przekroju elementu w odróżnieniu do np. przekrojów żelbetowych gdzie powierzchnia prętów zbrojeniowych jest pomijalnie mała. Elementy zespolone znajdują zastosowanie zarówno w budownictwie kubaturowym (płyty stropowe, podciągi, słupy) jak i w konstrukcjach mostowych (dźwigary mostów drogowych, a ostatnio również mostów kolejowych z tzw. korytem balastowym). W ujęciu klasycznym elementy zespolone składają się ze stali i betonu lub stali i żelbetu. Pozwalają one na wykorzystanie zalet poszczególnych materiałów składowych np. wytrzymałości betonu na ściskanie i wytrzymałości stali na rozciąganie. W efekcie przekroje elementów zespolonych mogą być kształtowane w sposób bardziej optymalny pod względem zużycia materiałów co prowadzi do redukcji kosztów i zwiększenia ich nośności. Beton jako materiał ogniotrwały może stanowić ochronę przeciwpożarową i antykorozyjną osadzonych w nim komponentów stalowych. Beton zmniejsza również smukłość elementów konstrukcyjnych chroniąc dźwigary i słupy stalowe przed utratą stateczności tj. wyboczeniem i zwichrzeniem. Zaletą stali jest natomiast lekkość, łatwość i szybkość montażu, którą można wykorzystać w przypadku konstrukcji zespolonych np. płyta żelbetowa na blasze fałdowej. Takie rozwiązanie pozwala skrócić czas wznoszenia konstrukcji i zmniejszyć straty wynikające z zamrożenia kapitału.
Typowymi przekrojami elementów zespolonych są:
1) Stalowe dwuteowe podciągi zespolone z płytą żelbetową w sposób umożliwiające włączenie do współpracy efektywnej szerokości płyty (rys. 1.1)
2) Płyta betonowa lub żelbetowa na blasze fałdowej, która stanowi dodatkowe zbrojenie rozciągane płyty oraz ułatwia i przyśpiesza wznoszenie konstrukcji. (rys. 1.2)
3) Słupy wykonane z obetonowanych dwuteowników z dodatkowym zbrojeniem (rys. 1.3)
4) Słupy z obetonowanym środnikiem dwuteownika i dodatkowym zbrojeniem (rys. 1.4)
5) Słupy z rur wypełnionych betonem z dodatkowym zbrojeniem (rys 1.5) – przyrost wytrzymałości betonu nawet o 30% ze względu na jego ograniczenie wewnątrz rury stalowej.



Rys. 1 Przykładowe przekroje zesolone [2]

Współpraca komponentów przekroju zespolonego najczęściej realizowana jest za pomocą

specjalnych łączników w taki sposób by zachowana była zgodność odkształceń na styku obu materiałów. Pozwala to na stosowanie hipotezy Bernoulliego tj. przekrój prosty i prostopadły do osi pręta przed deformacją pozostaje prosty i prostopadły po deformacji.

1.1 Cel Pracy

Elementy konstrukcyjne o przekrojach zespolonych lub żelbetowych ze względu na swoją geometrię, umiejscowienie w konstrukcji np. skrajne słupy, słupy pod dwukierunkowo zbrojonymi płytami, filary i pomosty mostów, jak również sposób przyłożenia obciążeń zewnętrznych są najczęściej rozpatrywane jako poddane jednoczesnemu działaniu siły osiowej N oraz dwukierunkowemu zginaniu Mx, My. Podstawowym celem niniejszej pracy będzie analiza odpowiedzi tego typu przekrojów zespolonych na zginanie ukośne z siłą osiową. Ocena nośności w stanie granicznym dokonywana jest poprzez konstruowanie trójwymiarowych powierzchni interakcji lub ich przekrojów nazywanych krzywymi interakcji. Możliwe jest również skonstruowanie krzywych moment-krzywizna (M-φ). Ze względu na możliwość występowania przekrojów zespolonych złożonych z wielu komponentów (stal, beton, zbrojenie, otwory etc.) i mających zróżnicowaną geometrię poszukiwanie takich wykresów jest zadaniem czasochłonnym i złożonym obliczeniowo. W związku z tym poszukuje się różnych metod zarówno analitycznego jak i numerycznego rozwiązania problemu. W niniejszej pracy poddamy głębszej analizie jedną z metod numerycznych zaproponowaną przez Vassilis K. Papanikolaou w pracy „Analysis of arbitraty composite sections in biaxial bending and axial load". Metoda ta poddana pewnym modyfikacją zostanie zaimplementowana w języku C/C++ w postaci programu umożliwiającego definiowanie dowolnie złożonego przekroju zespolonego oraz obliczanie i rysowanie dla tak zdefiniowanego przekroju powierzchni i krzywych interakcji. Ostatnim etapem pracy będzie weryfikacja poprawności działania programu z wykorzystaniem różnych benchmarków znalezionych w literaturze.

1.2 Zakres pracy

Praca dotyczy trzech głównych zagadnień. Pierwszym jest omówienie podstaw teoretycznych związanych z wyznaczaniem powierzchni i krzywych interakcji. Drugim analiza algorytmu przedstawionego przez V. K. Papanikolau oraz jego modyfikacji wraz z omówieniem sposobu implementacji w języku C/C++. Ostatnim zagadnieniem jest omówienie sposobu działania programu i walidacja jego poprawności przy użyciu benchmarków.

Praca składa się z następujących rozdziałów:

1. Podstawy teoretyczne analizy przekroju zespolonego – przedstawienie podstawowych założeń, definicji i wzorów oraz wymagań normowych. Omówienie czym są powierzchnie i krzywe interakcji.
2. Numeryczny zapis przekroju o dowolnej złożoności – omówienie zaproponowanej przez V. K. Papanikolau metody definiowania przekroju zespolonego w języku programowania zorientowanym obiektowo jakim jest C++, linearyzacja komponentów powierzchniowych.
3. Numeryczne wyznaczanie charakterystyk geometrycznych przekroju – wzory stosowane do wyznaczania momentów statycznych, pola powierzchni, momentów bezwładności oraz ich zapis za pomocą procedur numerycznych, generowanie możliwych przecięć komponentów powierzchniowych oraz odpowiednie dodawanie lub odejmowanie ich wpływu.
4. Numeryczny zapis związków konstytutywnych materiałów – omówienie sposobu implementacji związków konstytutywnych w języku programowania
5. Całkowanie numeryczne naprężeń po powierzchni przekroju – dobór odpowiedniego algorytmy jest kluczowym problemem rzutującym na szybkość i dokładność obliczeń.
6. Generowanie profili odkształceń granicznych – rozdział prezentuje sposoby sterowania profilem odkształceń w zakresie odkształceń granicznych. Profil ten stanowi podstawę do

otrzymania wynikowych sił wewnętrznych w procesie całkowania naprężeń.

7. Konstrukcja powierzchni i krzywych interakcji – ostatnim etapem implementacji programu jest generowanie wykresów powierzchni i krzywych interakcji na podstawie otrzymanego w wyniku stosowanych procedur numerycznych zbioru punktów w układzie współrzędnych Mx, My, N.
8. Program komputerowy – prezentacja możliwości utworzonej aplikacji wraz z omówieniem sposobu wprowadzania danych wejściowych i interpretacją danych wyjściowych.
9. Benchmarki – walidacja poprawności działania programu i zastosowanych w nim algorytmów na podstawie przykładów z literatury.

2. PODSTAWY TEORETYCZNE ANALIZY PRZEKROJU ZESPOLONEGO

Jak już wspomniano we wstępie przekroje zespolone są stosowane coraz powszechniej we współczesnym budownictwie zarówno kubaturowym jak i mostowym w związku z możliwością połączenia zalet materiałów składowych przy jednoczesnej eliminacji ich wad. Najczęściej wykorzystywanymi materiałami w konstrukcjach zespolonych są beton, stal konstrukcyjna i stal zbrojeniowa. Poniższe rozważania z powodzeniem można stosować również do przekrojów żelbetowych zakładając, że są one szczególnym przypadkiem przekrojów zespolonych. Ze względu na zróżnicowany charakter obciążeń działających na elementy zespolone, złożoną ich geometrię oraz umiejscowienie w konstrukcji poszukując ogólnych algorytmów wyznaczania nośności przekrojów zespolonych naturalnym wydaje się rozważanie takich przekrojów jako poddanych działaniu dwuosiowego zginania z jednoczesnym działaniem siły osiowej. Taki rodzaj obciążenia w przekroju zespolonym możemy spotkać np. w przypadku skrajnych słupów, pomostów mostów poddanych oddziaływaniom wiatru lub trzęsieniu ziemi. W literaturze możemy wyróżnić przypadki przekrojów złożonych tj. składających się z kilku komponentów wykonanych z tego samego materiału pomiędzy, którymi występuje współpraca zapewniona przez odpowiednie ich połączenie. Takie belki złożone z zapewnioną współpracą komponentów na powierzchni ich styku mają większą nośność niż belki w których nie zastosowano łączników. Można rozważać również tzw. wzmocnienia przekroju, których celem może być np. zwiększenie nośności istniejących elementów konstrukcyjnych. Przykładem przekrojów wzmacnianych są np. belki drewniane wzmocnione stalowymi ceownikami. W przypadku elementów konstrukcyjnych składających się z wielu komponentów istotne może być uwzględnienie naprężeń stycznych od sił rozwarstwiających na powierzchni kontaktu. Sytuacja ta ma miejsce gdy kierunek wektora momentu zginającego jest równoległy do powierzchni styku komponentów. W końcu największe korzyści można uzyskać w przypadku stosowania elementów zespolonych, które złożone są z komponentów wykonanych z różnych materiałów. W celu rozróżnienia właściwych przekrojów zespolonych od np. przekrojów żelbetowych wprowadza się dodatkowe ograniczenie mówiące, że w przypadku przekrojów zespolonych powierzchnia poszczególnych materiałów składowych musi być znacząca.

2.1 Założenia stosowane przy analizie nośności przekroju zespolonego

W niniejszych rozważaniach przyjmuje się spełnienie zasad technicznej teorii zginania. Ma tutaj zastosowanie hipoteza płaskich przekrojów Brenoulliego mówiąca, że przekroje poprzeczne pręta płaskie i prostopadłe do osi pręta przed odkształceniem pozostają po deformacji płaskie i prostopadłe do ugiętej osi pręta.

$$\epsilon_x = \epsilon_0 + \phi \cdot \eta \quad (1)$$

gdzie:

ε_0 – odkształcenia odpowiadające współrzędnej $\eta = 0$ tj. początkowi lokalnego centralnego układu współrzędnych

ε_x - odkształcenia na wysokości przekroju odpowiadające współrzędnej η

η – odległość rozpatrywanego punktu przekroju od osi $\eta = 0$

φ – krzywizna, kąt nachylenia profilu odkształceń

Wzór ten oznacza, że odkształcenia zmieniają się liniowo na wysokości całego przekroju, jest to możliwe dzięki występującej na styku dwóch materiałów zgodności odkształceń $\varepsilon_{x1} = \varepsilon_{x2}$.
Ponadto przyjmuje się jednoosiowy stan naprężenia pomijając naprężenia styczne.

$$\sigma_x >> \sigma_y, \sigma_z \quad (2)$$

Związki konstytutywne dla materiałów komponentów składowych przekroju będą przyjmowane zgodnie z normą europejską dla konstrukcji zespolonych PN-EN 1994 pomimo iż implementowany przez nas algorytm będzie na tyle ogólny, że będzie pozwalał na stosowanie zależności naprężenie – odkształcenie w postaci dowolnej funkcji odcinkami wielomianowej.

Nośność analizowanego przekroju zespolonego będzie ustalana poprzez skonstruowanie odpowiednich krzywych i powierzchni interakcji.

2.2 Sprowadzone charakterystyki geometryczne oraz warunki równoważności sił wewnętrznych i zewnętrznych w układzie ważonym dla przekrojów zespolonych

W części praktycznej pracy omawiającej sposób implementacji programu istotnym etapem obliczeń będzie znajdowanie geometrycznego środka ciężkości przekroju zespolonego. Będzie to środek ciężkości C w tzw. osiach ważonych (y*, z*) do którego wyznaczenia potrzebne są ważone charakterystyki materiałowo-geometryczne tj. ważone pole przekroju A*, ważone momenty statyczne Sy*, Sz*. Dodatkową charakterystyką przydatną w obliczeniach analitycznych nośności przekroju zespolonego są ważone momenty bezwładności Jy*, Jz*. Dla przekroju zespolonego złożonego z n komponentów sprowadzone charakterystyki geometryczne mogą być wyrażone następującymi wzorami.

Ważone pole przekroju:

$$A^* = \sum_{i=1}^{k} n_i \cdot A_i \quad (3)$$

Ważone momenty statyczne:

$$S_y^* = \sum_{i=1}^{k} n_i \cdot S_{yi} \quad (4)$$

$$S_z^* = \sum_{i=1}^{k} n_i \cdot S_{zi} \quad (5)$$

Ważone momenty bezwładności:

$$I_y^* = \sum_{i=1}^{k} n_i \cdot I_{yi}^* \quad (6)$$

$$I_z^* = \sum_{i=1}^{k} n_i \cdot I_{zi}^* \quad (7)$$

gdzie n_i oznacza wagę wyrażoną jako: $n_i = \frac{E_i}{E_1}$ natomiast gwiazdka przy I_{yi}^*, I_{zi}^* oznacza, że momenty bezwładności i-tego komponentu przekroju są liczone względem osi ważonych, odpowiednio y^*, z^* . Położenie tych osi może być wyrażone za pomocą poniższych wzorów:

$$y^* = \frac{S_z^*}{A^*} \quad (8) \qquad z^* = \frac{S_y^*}{A^*} \quad (9)$$

Powyższe wzory wyrażające sprowadzone charakterystyki przekrojów zespolonych umożliwiają dokonywanie obliczeń w układzie osi ważonych $y^* C z^*$, a co za tym idzie możliwość potraktowania przekrojów materiałowo niejednorodnych tak jakby były przekrojami jednorodnymi. Oznacza to separację stanu tarczowego od stanu giętnego tj. traktowanie w sposób niezależny wydłużenia lub skrócenia osi pręta wywołanego jedynie przez siłę osiową oraz ugięcia osi pręta wywołanego tylko momentem zginającym. W rzeczywistości w przekrojach zespolonych mamy do czynienia ze sprzężeniem stanu tarczowego i giętnego o czym można się przekonać analizując równania równowagi sił wewnętrznych i zewnętrznych.

$$N = \iint_A \sigma_x \, dA = \sum_{i=1}^{k} \iint_{A_i} \sigma_{x,i} \, dA_i \quad (10)$$

$$M_y = \iint_A \sigma_x \cdot z \, dA = \sum_{i=1}^{k} \iint_{A_i} \sigma_{x,i} \cdot z \, dA_i \quad (11)$$

$$-M_z = \iint_A \sigma_x \cdot y \, dA = \sum_{i=1}^{k} \iint_{A_i} \sigma_{x,i} \cdot y \, dA_i \quad (12)$$

Podstawiając do powyższych wzorów odpowiednio zależności:

$$\sigma_{x,i} = E_i \cdot \epsilon_x \quad (13)$$

$$\epsilon_x = \epsilon_0 + \phi_y \cdot z + \phi_z \cdot y \quad (14)$$

$$\sigma_{x,i} = E_i \cdot (\epsilon_0 + \phi_y \cdot z + \phi_z \cdot y) \quad (15)$$

Gdzie wzór (14) otrzymano na podstawie liniowego rozkładu odkształceń w przekroju ze wzoru (1) po zapisaniu go w globalnym centralnym układzie współrzędnych. Oznacza to, że krzywizna od zginania ukośnego w lokalnym układzie współrzędnych $\eta C \xi$ została rozłożona na dwie składowe krzywizny ϕ_y, ϕ_z względem osi układu globalnego yCz, odpowiadające działaniu odpowiednio momentów składowych M_y oraz M_z .

$$\phi = \sqrt{\phi_y^2 + \phi_z^2} \quad (16)$$

W efekcie warunki równoważności (10), (11), (12) możemy zapisać w następujący sposób:

$$N = \epsilon_0 \cdot \left(\sum_{i=1}^{k} E_i \cdot A_i\right) + \phi_y \cdot \left(\sum_{i=1}^{k} E_i \cdot S_{y,i}\right) + \phi_z \cdot \left(\sum_{i=1}^{k} E_i \cdot S_{z,i}\right) \quad (17)$$

$$M_y = \epsilon_0 \cdot \left(\sum_{i=1}^{k} E_i \cdot S_{y,i}\right) + \phi_y \cdot \left(\sum_{i=1}^{k} E_i \cdot I_{y,i}\right) + \phi_z \cdot \left(\sum_{i=1}^{k} E_i \cdot I_{yz,i}\right) \quad (18)$$

$$-M_z = \epsilon_0 \cdot \left(\sum_{i=1}^{k} E_i \cdot S_{z,i}\right) + \phi_y \cdot \left(\sum_{i=1}^{k} E_i \cdot I_{yz,i}\right) + \phi_z \cdot \left(\sum_{i=1}^{k} E_i \cdot I_{z,i}\right) \quad (19)$$

W powyższych wzorach $S_{y,i}, S_{z,i}, I_{y,i}, I_{z,i}, I_{yz,i}$ są odpowiednio momentami statycznymi względem osi y, względem osi z oraz momentami bezwładności względem osi y, względem osi z oraz momentem dewiacji obliczonymi dla i-tego komponentu przekroju zespolonego względem globalnego układu centralnego. Z wzoru (16) wynika, że siła osiowa N powoduje zarówno wydłużenie/skrócenie osi jak również jej ugięcia w kierunkach osi y-y oraz z-z.
Z wzorów (17) i (18) wynika natomiast, że momenty zginające M_y oraz M_z powodują oprócz ugięcia również wydłużenie/skrócenie osi pręta. Gdyby w powyższych wzorach założyć, że przekrój jest wykonany z materiału jednorodnego tj. $E_i = const\, E$ można by wyłączyć stałą E przed poszczególne sumy i w efekcie otrzymać zerowanie składników

$$\sum_{i=1}^{k} S_{y,i} = 0, \sum_{i=1}^{k} S_{z,i} = 0 \quad (20)$$

jako, że momenty statyczne przekroju względem osi centralnych są równe 0. Końcowym efektem byłoby otrzymanie w przypadku siły osiowej N jedynie stanu tarczowego natomiast w przypadku momentów zginających M jedynie stanu giętnego.

Aby dokonać rozdziału stanów tarczowego i giętnego w przekrojach zespolonych należy znaleźć położenie tzw. sprowadzonych osi ciężkości y^*, z^* . Można tego dokonać z poniższych warunków:

$$\sum_{i=1}^{k} E_i \cdot S^*_{y,i} = 0 \quad (21)$$

$$\sum_{i=1}^{k} E_i \cdot S^*_{z,i} = 0 \quad (22)$$

Gwiazdka w powyższych równaniach przy momentach statycznych i-tego komponentu przekroju oznacza, że są one liczone względem poszukiwanych sprowadzonych osi ciężkości, odpowiednio y^* i z^* . W miejsce momentów statycznych $S^*_{y,i}, S^*_{z,i}$ należy wstawić zależności z twierdzenia Steinera wyrażające przejście z początkowego układu w którym wyznaczane są momenty statyczne i-tego komponentu przekroju do układu osi ważonych.

$$S^*_{y,i} = S_{y,i} - A_i \cdot z^* \quad (23)$$

$$S^*_{z,i} = S_{z,i} - A_i \cdot y^* \quad (24)$$

W efekcie po przekształceniach otrzymamy wzory na ważone osie ciężkości wyrażone względem początkowo przyjętego układu współrzędnych:

$$z^* = \frac{\sum_{i=1}^{k} E_i \cdot S_{y,i}}{\sum_{i=1}^{k} E_i \cdot A_i} = \frac{\sum_{i=1}^{k} \frac{E_i}{E_1} \cdot S_{y,i}}{\sum_{i=1}^{k} \frac{E_i}{E_1} \cdot A_i} = \frac{\sum_{i=1}^{k} n_i \cdot S_{y,i}}{\sum_{i=1}^{k} n_i \cdot A_i} \quad (25)$$

$$y^* = \frac{\sum_{i=1}^{k} E_i \cdot S_{z,i}}{\sum_{i=1}^{k} E_i \cdot A_i} = \frac{\sum_{i=1}^{k} \frac{E_i}{E_1} \cdot S_{z,i}}{\sum_{i=1}^{k} \frac{E_i}{E_1} \cdot A_i} = \frac{\sum_{i=1}^{k} n_i \cdot S_{z,i}}{\sum_{i=1}^{k} n_i \cdot A_i} \quad (26)$$

Warto zauważyć, że w powyższych wzorach (25), (26) otrzymano w licznikach zdefiniowane uprzednio w (4), (5) ważone momenty statyczne, a w mianownikach zdefiniowaną w (3) ważoną powierzchnię przekroju.

Mając sprowadzone charakterystyki geometryczne przekroju zespolonego możemy zapisać równania równoważności sił wewnętrznych i zewnętrznych w układzie ważonym $y^* C z^*$.

$$N = \epsilon_0 \cdot (\sum_{i=1}^{k} E_i \cdot A_i) + \phi_y \cdot (\sum_{i=1}^{k} E_i \cdot S^*_{y,i}) + \phi_z \cdot (\sum_{i=1}^{k} E_i \cdot S^*_{z,i}) = \epsilon_0 \cdot (\sum_{i=1}^{k} E_i \cdot A_i) = \epsilon_0 \cdot E_1 \cdot (\sum_{i=1}^{k} n_i \cdot A_i) = \epsilon_0 \cdot E_1 \cdot A^* \quad (27)$$

$$M^*_y = \epsilon_0 \cdot (\sum_{i=1}^{k} E_i \cdot S^*_{y,i}) + \phi_y \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{y,i}) + \phi_z \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{yz,i}) = \phi_y \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{y,i}) + \phi_z \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{yz,i}) \quad (28)$$

W przypadku układu głównych centralnych osi bezwładności moment dewiacji I^*_{yz} zeruje się, a momenty bezwładności są ekstremalne. Stąd formułę (28) można uprościć do następującej postaci:

$$M^*_y = \phi_y \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{y,i}) + \phi_z \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{yz,i}) = \phi_y \cdot E_1 \cdot (\sum_{i=1}^{k} n_i \cdot I^*_{y,i}) = \phi_y \cdot E_1 \cdot I^*_y \quad (29)$$

Analogiczna sytuacja ma miejsce w przypadku momentu względem osi z-z M_z

$$-M^*_z = \phi_y \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{yz,i}) + \phi_z \cdot (\sum_{i=1}^{k} E_i \cdot I^*_{z,i}) = \phi_z \cdot E_1 \cdot (\sum_{i=1}^{k} n_i \cdot I^*_{z,i}) = \phi_z \cdot E_1 \cdot I^*_z \quad (30)$$

Przekształcając powyższe warunki równoważności (27), (29), (30) można wyznaczyć wzory na parametry rozkładu odkształceń:

$$\epsilon_0 = \frac{N}{E_1 \cdot A^*} \quad (31)$$

$$\phi_y = \frac{M^*_y}{E_1 \cdot I^*_y} \quad (32)$$

$$\phi_z = \frac{-M^*_z}{E_1 \cdot I^*_z} \quad (33)$$

Związki te są podstawą do sformułowania wzorów na odkształcenia liniowe ϵ_x , naprężenia w poszczególnych częściach przekroju $\sigma_{x,i}$ oraz równania osi obojętnej:

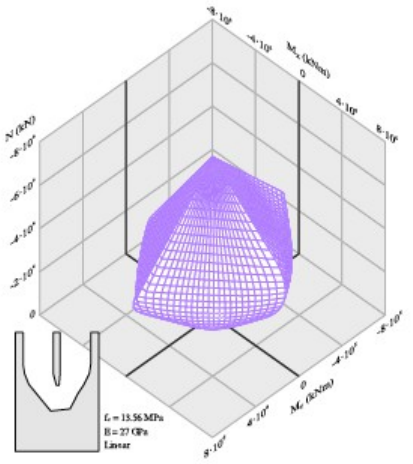
$$\epsilon_x = \frac{1}{E_1} \cdot \left(\frac{N}{A^*} + \frac{M^*_y}{I^*_y} \cdot z - \frac{M^*_z}{I^*_z} \cdot y \right) \quad (34)$$

$$\sigma_{x,i} = E_i \cdot \epsilon_x = n_i \cdot \left(\frac{N}{A^*} + \frac{M_y^*}{I_y^*} \cdot z - \frac{M_z^*}{I_z^*} \cdot y \right) \quad (35)$$

$$z_{neutral} = \left(\frac{M_z^*}{I_z^*} \cdot y - \frac{N}{A^*} \right) \cdot \frac{I_y^*}{M_y^*} \quad (36)$$

2.3 Powierzchnie i krzywe interakcji

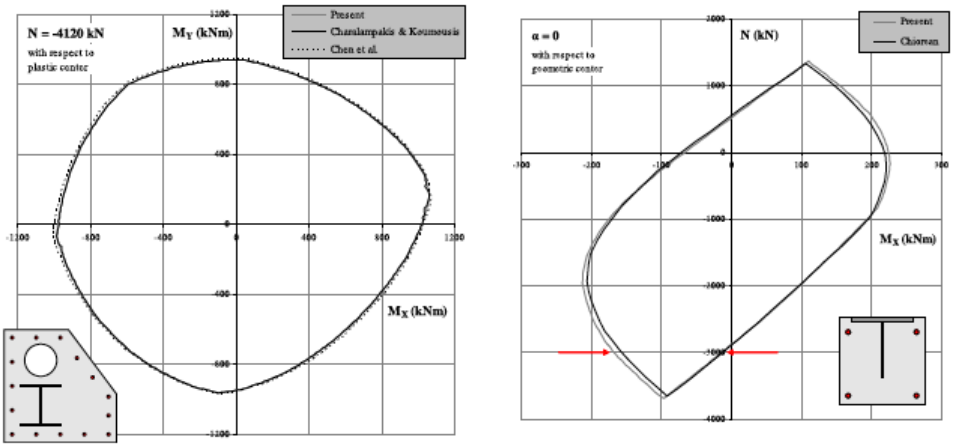
Stan granicznej nośności przekroju zespolonego lub żelbetowego poddanego działaniu dwuosiowego zginania M_y, M_z i siły osiowej N może być zobrazowany w postaci powierzchni interakcji (ang. interaction surfaces) lub krzywych interakcji (ang. interaction curves). Powierzchnie interakcji konstruowane są w trójwymiarowej przestrzeni (M_x, M_y, N) na podstawie zbioru punktów otrzymanego w wyniku zastosowania procedur całkowania naprężeń w przekroju tj. z warunków równoważności (10), (11), (12). Powierzchnie interakcji nazywane są również powierzchniami zniszczenia (ang. failure surfaces) jako, że przedstawiają obwiednie nośności przekroju. Oznacza to, że punkty N, M_x, M_y leżące na tej powierzchni stanowią graniczne wartości sił zewnętrznych jakie przekrój jest w stanie przenieść. Powierzchnia interakcji tworzy wypukłą bryłę ograniczającą zbiór punktów możliwych do przeniesienia przez przekrój. Wszystkie punkty leżące na zewnątrz tej bryły reprezentują kombinację sił N, M_x, M_y prowadzącą do zniszczenia przekroju. Punktem wyjścia do otrzymania powierzchni interakcji jest wyznaczenie zbioru tworzących ją punktów. Zbiór taki można otrzymać rozpatrując kolejno graniczne profile odkształceń. Każdemu profilowi odkształceń odpowiada dany rozkład naprężeń w przekroju. Stosując wspomniane już warunki równoważności (10), (11), (12) tj. całkując naprężenia w przekroju otrzymujemy odpowiednie siły N, M_x, M_y będące współrzędnymi punktów we wspomnianej przestrzeni trójwymiarowej (M_x, M_y, N) .



Rys. 3 Przykładowa powierzchnia interakcji [1]

Krzywe interakcji stanowią natomiast przekroje odpowiednimi płaszczyznami otrzymanej powierzchni interakcji. W literaturze rozpatruje się głównie krzywe interakcji $M_x - M_y$ otrzymane dla stałej wartości siły osiowej N oraz krzywe interakcji N-M uzyskiwane dla ustalonego kąta α . Kąt α jest to kąt pomiędzy wypadkowym momentem zginającym

$M=\sqrt{M_x^2+M_y^2}$, a dodatnią osią OM_x . Przyjmując wartość kąta $\alpha=0^\circ$ otrzymamy krzywą interakcji $N-M_x$ dla $M_y=0\,kNm$, natomiast dla wartości kąta $\alpha=90^\circ$ krzywą interakcji $N-M_y$ przy $M_x=0\,kNm$. Cechą charakterystyczną krzywych interakcji jest ich wypukłość. Ponadto istnieje możliwość skonstruowania krzywych interakcji zarówno przy założeniu sprężystego rozkładu naprężeń w przekroju jak i przy uwzględnieniu zakresu sprężysto-plastycznego materiałów składowych. W tym drugim przypadku tj. dla stanu granicznej nośności plastycznej otrzymane krzywe nazywa się krzywymi nośności.



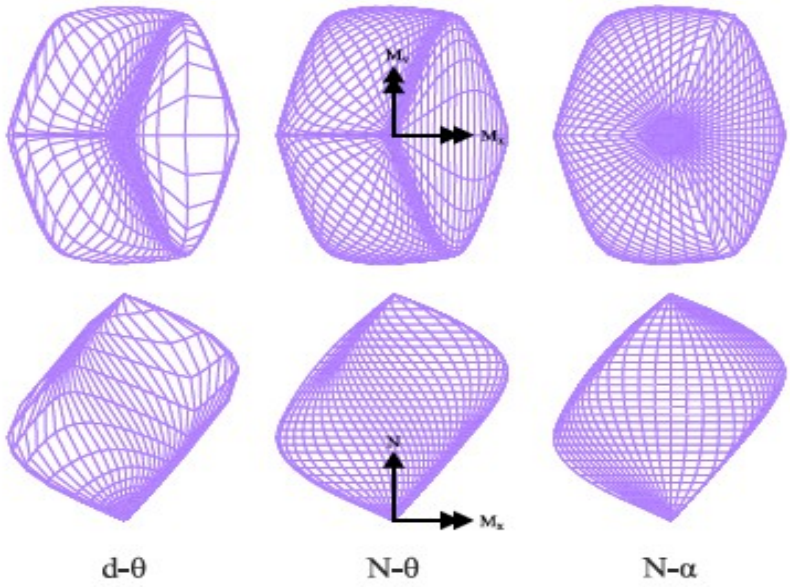
Rys. 4 Przykładowa krzywa interakcji Mx-My oraz N-Mx [1]

Papanikolau w [1] rozpatrzył trzy podejścia do generowania profili odkształceń w celu wyznaczenia punktów N, M_x, M_y potrzebnych do skonstruowania powierzchni i krzywych interakcji.

- Sterowanie $d-\theta$ - w tym przypadku kolejne profile odkształceń generowane są iteracyjnie poprzez obrót osi obojętnej o kąt θ w zakresie $[0^\circ, 360^\circ]$ oraz zmianę odległości osi obojętnej od początku lokalnego układu współrzędnych d w zakresie $[-\infty, +\infty]$ Punkty N, M_x, M_y otrzymuje się w wyniku całkowania rozkładów naprężeń w przekroju odpowiadających wygenerowanym profilom odkształceń. Takie podejście daje jednak nieregularną siatkę punktów w przestrzeni. Podobne do tego podejścia jest sterowanie profilem odkształceń w zakresie dopuszczalnych odkształceń granicznych poprzez iterowanie kątem obrotu θ oraz skrajnymi odkształceniami ϵ_{upper} w zakresie $[\epsilon_{tu}, \epsilon_{cu}]$ oraz ϵ_{lower} w zakresie $[\epsilon_{tu}, \epsilon_{cu-co}]$ gdzie ϵ_{cu-co} jest zależne od dopuszczalnego położenia $\epsilon_{cu}, \epsilon_{co}$. Metoda ta będzie wykorzystana w implementacji programu i dokładniej omówiona w części praktycznej tej pracy.
- Sterowanie $N-\theta$ - w tym przypadku profile odkształceń generowane są sterując odpowiednio kątem obrotu osi obojętnej θ (lokalnego układu współrzędnych) oraz wartością siły osiowej N . Dla ustalonych wartości tych dwóch parametrów uzyskiwane są odległości osi obojętnej od początku układu współrzędnych d. Stosuje się tutaj metodę poszukiwania pierwiastka funkcji nieliniowej Brent'a. Wartość d poszukuje się w sposób iteracyjny zakładając pewne początkowe d_0 . Dla tej wartości oraz ustalanego kąta θ w wyniku całkowania odpowiadającego im rozkładu naprężeń uzyskiwana jest pewna wartość N_0 . Następnie porównuje się ją z ustaloną wartością N i w zależności od wyniku porównania określany jest kierunek poszukiwania pierwiastka d jako pozytywny lub negatywny. Kolejne wartości d , które powinny zbliżać się do tej odpowiadającej ustalonemu N uzyskiwane są na podstawie przyjętego kierunku poszukiwania i założonego kroku iteracyjnego (zalecana iteracja wykładnicza) aż do momentu zmiany

kierunku. W efekcie otrzymuje się przedział $[d_A, d_B]$ tzw. przedział Brent'a w którym znajduje się szukana wartość d odpowiadająca ustalonej wartości N . Pozwala to na zawężenie zakresu iteracji i osiągnięcie zbieżności poszukiwań. Prezentowane podejście umożliwia efektywne znalezienie punktów krzywej interakcji $M_x - M_y$ dla ustalonej wartości siły osiowej N . W przypadku konstruowania powierzchni zniszczenia uzyskiwana siatka punktów jest bardziej regularna niż w sterowaniu $d-\theta$ tj. południki są równoodległe i leżą w jednej płaszczyźnie odpowiadającej wartości siły osiowej N .

- Sterowanie $N-\alpha$ - profile odkształceń są generowane poprzez sterowanie siłą osiową N oraz kątem nachylenia momentu M: α . Stosuje się tutaj podobnie jak w podpunkcie drugim poszukiwanie metodą Brent'a wartości parametrów (d, θ) definiujących jednoznacznie profil odkształceń dla ustalonych wartości N, α . Ze względu na potrzebę znalezienia dwóch parametrów stosuje się podwójne zagnieżdżenie metody Brent'a tj. w zewnętrznej pętli iterujemy po wartościach poszukiwanego kąta θ przyrównując wynik całkowania do ustalonego kąta α , w wewnętrznej pętli iterujemy po wartościach poszukiwanej odległości osi obojętnej d , mając bieżącą wartość θ i przyrównując wynik całkowania do ustalonej siły osiowej N . Ze względu na sterowanie zarówno siłą osiową N jak i kątem nachylenia momentu α otrzymana siatka punktów jest najbardziej równomiernie rozłożona, jednocześnie jest to metoda najbardziej kosztowna obliczeniowo. Zarówno południki jak i równoleżniki trójwymiarowej powierzchni zniszczenia są równoodległe i leżą w jednej płaszczyźnie odpowiadającej odpowiednio sile osiowej i kątowi nachylenia momentu.



Rys. 5 Sposoby generowania siatki punktów dla powierzchni interakcji [1]

Warto zauważyć, że środki poszczególnych krzywych interakcji $M_x - M_y$ leżą na jednej prostej łączącej wierzchołki odpowiadające granicznym siłą osiowym ściskającej N_{min} i rozciągającej N_{max} , która w przypadku przekrojów symetrycznych jest pionowa i przechodzi przez punkty (0,0) płaszczyzn $M_x - M_y$. Powierzchnia zniszczenia w przypadku takich przekrojów jest powierzchnią obrotową względem tej pionowej prostej. W przypadku przekrojów niesymetrycznych prosta ta jest nachylona pod pewnym kątem, a w łączonych wierzchołkach z granicznymi wartościami sił osiowych momenty M_x, M_y nie są zerowe jak to ma miejsce w przypadku przekrojów symetrycznych.

2.4 Związki konstytutywne dla materiałów wg PN-EN 1994

W celu przeprowadzenia poprawnej analizy nośności przekroju zespolonego należy stosować realne modele pracy materiałów z jakich się on składa. W realizowanym programie predefiniowane zostały związki konstytutywne $\sigma-\epsilon$ zgodne z zalecaniami normy europejskiej PN-EN 1994. [3] Aczkolwiek implementowany algorytm pozwala na stosowanie dowolnych zależności naprężenie-odkształcenie w postaci funkcji odcinkami wielomianowej, które pozwalają na uzyskiwanie dokładnych wyników przy wykorzystaniu całkowania numerycznego Gaussa. Zastosowanie zależności częściowo wykładniczych byłoby możliwe po wprowadzeniu modyfikacji proponowanej przez Papanikolau'a [1]. Zgodnie z rozdziałem 3 normy przewidzianej do projektowania konstrukcji zespolonych PN-EN 1994-1-1 właściwości betonu i stali zbrojeniowej należy przyjmować jak w normie dla konstrukcji żelbetowych PN-EN 1992-1-1, przy czym obliczeniowe wartości modułów sprężystości stali zbrojeniowej E_s można przyjmować według PN-EN 1993-1-1/3.2.6 jak dla stali konstrukcyjnej. Analogicznie zalecane jest przyjmowanie właściwości materiałowych stali konstrukcyjnej zgodnie z odpowiednią normą przewidzianą dla konstrukcji stalowych PN-EN 1993-1-1.

Rozpatrując właściwości poszczególnych materiałów należy posługiwać się trzema wartościami granicznych odkształceń $\epsilon_{cu}, \epsilon_{co}, \epsilon_{tu}$, które są odpowiednio odkształceniami przy granicznej wytrzymałości na ściskanie, wytrzymałości na ściskanie przy czystym ściskaniu oraz granicznej wytrzymałości na rozciąganie. By poprawnie zdefiniować związki konstytutywne potrzebna jest znajomości tych wartości oraz odpowiadających im wartości wytrzymałości tj. naprężeń dla materiału danej klasy lub gatunku. Niektóre z tych parametrów odkształceń mogą być pominięte w zależności od rodzaju materiału, np. ϵ_{co} dla stali lub wytrzymałość na rozciąganie dla betonu.

Właściwości betonu są determinowane przez jego klasę, która bezpośrednio związana jest z wytrzymałością charakterystyczną betonu na ściskanie określaną jako 5% wytrzymałość walcowa f_{ck} lub wytrzymałość kostkowa $f_{ck,cube}$ według EN 206-1. Klasy wytrzymałości betonu podane w PN-EN 1992-1-1 opierają się na wytrzymałości walcowej określonej po 28 dniach. Są one oznacza według schematu C20/25 gdzie pierwsza wartość oznacza wytrzymałość walcową na ściskanie $f_{ck}=20\text{MPa}$, druga wytrzymałość kostkową na ściskanie $f_{ck,cube}=25\text{MPa}$.
Innymi ważnymi charakterystykami materiałowymi dla betonu są średnia wytrzymałość na ściskanie f_{cm}, wytrzymałość charakterystyczna na rozciąganie f_{ctk}, moduł sprężystości betonu E_{cm}.

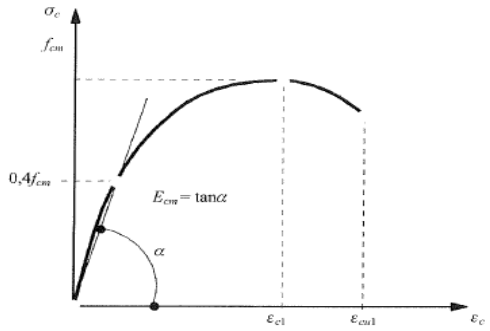
$$f_{cm}=f_{ck}+8 \quad [f_{ck}\,w\,MPa] \quad (37)$$

$$E_{cm}=22\cdot(0.1\cdot f_{cm})^{0.3} \quad [f_{cm}\,w\,MPa] \quad (38)$$

Podczas projektowania przekrojów z betonu zgodnie z PN-EN 1992 należy przyjmować wartości obliczeniowe wytrzymałości na ściskanie zgodnie z poniższą zależnością.

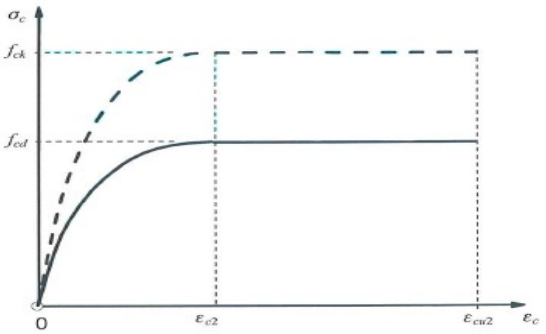
$$f_{cd}=\alpha_{cc}\cdot\frac{f_{ck}}{\gamma_c} \quad (39)$$

Gdzie α_{cc} jest współczynnikiem korekcyjnym którego zalecana wartość to 1.0, natomiast γ_c to częściowy współczynnik bezpieczeństwa, który w trwałych i przejściowych sytuacjach obliczeniowych zgodnie z PN-EN 1991-1-1 [9] przyjmuje wartość 1.5. Norma PN-EN 1992-1-1 podaje w Tablicy 3.1 wartości odkształceń granicznych w zależności od klasy betonu dla trzech modeli pracy materiału. Wartości $\epsilon_{c1}, \epsilon_{cu1}$ dla zależności naprężenie-odkształcenie do nieliniowej analizy konstrukcji (Rys. 6)

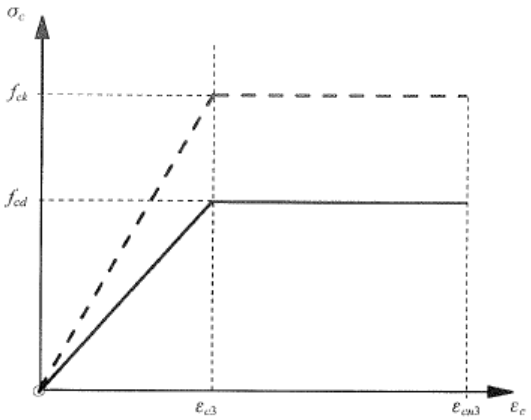


Rys. 6 Model zależności $\sigma - \epsilon$ do analizy nieliniowej [4]

Wartości $\epsilon_{c2}, \epsilon_{cu2}$ dla modelu parabola-prostokąt (Rys. 7) oraz $\epsilon_{c3}, \epsilon_{cu3}$ dla bilinearnej zależności naprężenie-odkształcenie (Rys. 8)



Rys. 7 Model zależności $\sigma - \epsilon$ parabola-prostokąt [4]



Rys. 8 Bilinearna zależność $\sigma - \epsilon$ [4]

Norma PN-EN 1992-1-1 dopuszcza również stosowanie prostokątnego rozkładu naprężeń dla analizy uproszczonej przekrojów. W przypadku realizowanego programu wykorzystano zależność paraboliczno-liniową (Rys. 7) oraz wartości granicznych odkształceń $\epsilon_{c2}, \epsilon_{cu2}$. Warto zauważyć iż wszystkie powyższe modele pomijają wytrzymałość betonu na rozciąganie. Odkształcenia graniczne w modelu paraboliczno-liniowym dla betonów klasy C12/15 – C50/60 wynoszą $\epsilon_{c2} = 0.002$ (dla czystego ściskania) oraz $\epsilon_{cu2} = 0.0035$ (dla granicznej wytrzymałości na ściskanie), natomiast dla betonu o $f_{ck} > 50\text{MPa}$ wyrażone są zależnościami:

$$\epsilon_{c2} = 2.0 + 0.085 \cdot (f_{ck} - 50)^{0.53} \ [\,^0/_{00}] \quad (40)$$

$$\epsilon_{cu2} = 2.6 + 35 \cdot [0.01 \cdot (90 - f_{ck})]^4 \ [\,^0/_{00}] \quad (41)$$

Związek konstytutywny dla betonu (Rys. 7) wyrażony jest poprzez funkcję złożoną $\sigma = f(\epsilon)$, którą można wyrazić następującymi wzorami:

$$\sigma_c = 0 \quad dla \quad \epsilon_c \leq 0 \quad (42)$$

$$\sigma_c = f_{cd} \cdot [1 - (1 - \frac{\epsilon_c}{\epsilon_{c2}})^n] \quad dla \quad 0 \leq \epsilon_c \leq \epsilon_{c2} \quad (43)$$

$$\sigma_c = f_{cd} \quad dla \quad \epsilon_{c2} \leq \epsilon_c \leq \epsilon_{cu2} \quad (44)$$

W odniesieniu do stali zbrojeniowej norma PN-EN 1992-1-1 dopuszcza stosowanie stali zbrojeniowej żebrowanej o charakterystycznej graniczy plastyczności 400-600MPa. Dobre parametry wytrzymałościowe stali nie są jedynak jedynym istotnym kryterium równie ważna jest odpowiednia ciągliwość stali. Ciągliwość stali oznacza możliwość osiągania znaczących przyrostów odkształceń przy niewielkim wzroście naprężeń po przekroczeniu granicy plastyczności. Znaczenie ciągliwości zostało podkreślone w normie europejskiej poprzez podział stali zbrojeniowej na trzy klasy A, B, C właśnie ze względu na ten parametr wyrażony poniższym wzorem:

$$k = \frac{f_{tk}}{f_{yk}} \quad (45)$$

Poniższa tabela obrazuje ten podział oraz podaje przykładowe gatunki stali spełniające podane kryteria, które predefiniowano w realizowanym programie.

Właściwość zbrojenia	Klasy stali		
	A	B	C
Charakterystyczna granica plastyczności $f_{yk}[MPa]$	400-600		
Minimalna ciągliwość k	$k \geq 1.05$	$k \geq 1.08$	$k \geq 1.35$
Charakterystyczne odkształcenia graniczne przy rozciąganiu $\epsilon_{uk}[\%]$	$\epsilon_{uk} \geq 2.5$	$\epsilon_{uk} \geq 5.0$	$\epsilon_{uk} \geq 7.5$
Gatunki stali spełniające podane kryteria	St3SY-b-500	RB 500W Bst500S	B500SP [Epstal]

Tab. 1 Klasyfikacja stali zbrojeniowej według PN-EN 1992-1-1 [10]

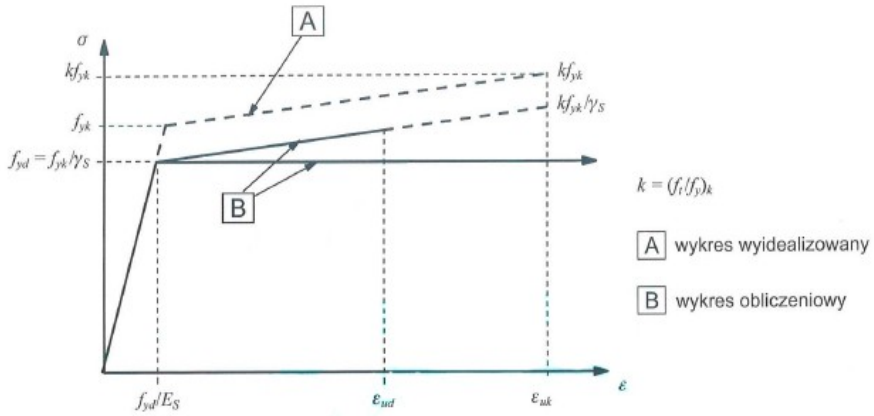
Do projektowania przekrojów zawierających stal zbrojeniową należy stosować wartości obliczeniowe charakterystyk materiałowych tj. wartości charakterystyczne podzielone przez częściowy współczynnik bezpieczeństwa $\gamma_s = 1.15$ dla trwałych i przejściowych sytuacji obliczeniowych.

$$f_{yd} = \frac{f_{yk}}{\gamma_s} \quad (46)$$

Norma PN-EN 1992-1-1 zaleca stosowanie alternatywnie jednego z dwóch wyidealizowanych modeli bilinearnych naprężenie-odkształcenie dla stali zbrojeniowej (Rys. 10). Istnieje możliwość

przyjęcia poziomej górnej gałęzi wykresu gdzie odkształcenie graniczne wynosi $\epsilon_{ud}=0.9\cdot\epsilon_{uk}$, a maksymalne naprężenie jest równe obliczeniowej granicy plastyczności f_{yd} (model sprężysto-plastyczny Prandtla) lub przyjęcie pochylonej górnej gałęzi, gdzie maksymalne naprężenie jest równe $\frac{k\cdot f_{yk}}{\gamma_s}$ (model sprężysto-plastyczny ze wzmocnieniem).

W realizowanym programie przyjęto wariant zależności $\sigma-\epsilon$ ze wzmocnieniem. Warto zauważyć iż przedstawiony w normie wykres uwzględnia jedynie fragment reprezentujący zachowanie się stali zbrojeniowej poddanej rozciąganiu. W celu uzyskania pełnego związku konstytutywnego dla stali zbrojeniowej należy odbić wykres normowy symetrycznie względem początku układu współrzędnych.

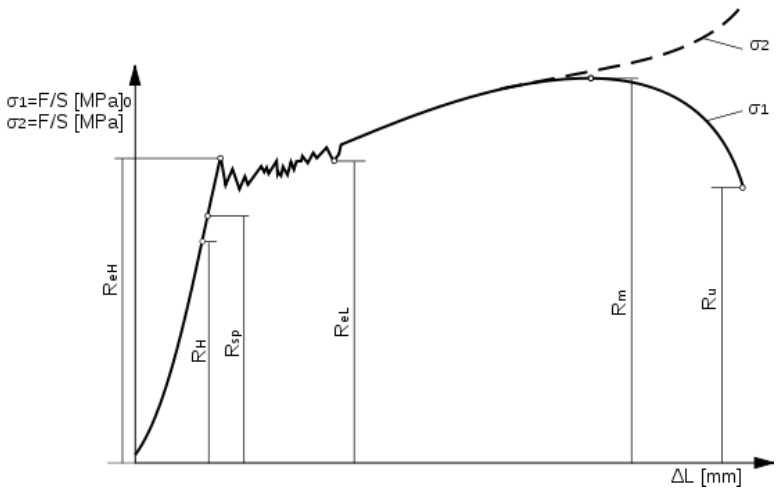


Rys. 10 Bilinearna zależność $\sigma-\epsilon$ dla stali zbrojeniowej [4]

Norma europejska do projektowania konstrukcji stalowych PN-EN 1993-1-1 określa wymagania odnośnie stosowanej w elementach zespolonych stali konstrukcyjnej. Powinna ona posiadać minimalną ciągliwość $f_u/f_y\geq 1.10$, wydłużenie przy zniszczeniu ≥ 15 , odkształcenie po osiągnięciu granicznej wytrzymałości na rozciąganie $\epsilon_u\geq 15\cdot\epsilon_y$, gdzie ϵ_y to odkształcenie przy uplastycznieniu:

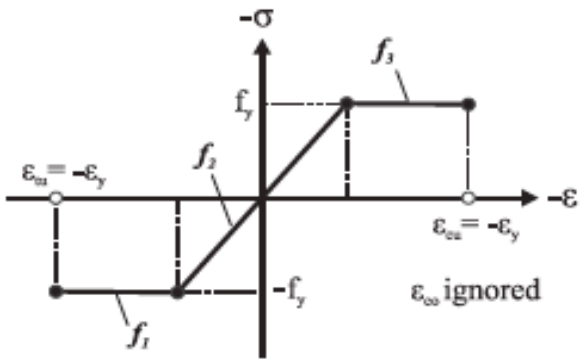
$$\epsilon_y=\frac{f_y}{E} \quad (47)$$

Zależność naprężenie-odkształcenie dla stali z wyraźną granicą plastyczności wygląda jak na Rys. 11.



Rys. 11 Zależność $\sigma - \epsilon$ dla stali z wyraźną granicą plastyczności [11]

Do analizy konstrukcji można przyjąć uproszczony model pracy stali np. model sprężysto-plastyczny Prandtla. (Rys. 12) Istotnymi parametrami na takim wyidealizowanym bilinearnym wykresie są granica plastyczności f_y odnosząca się do minimalnej górnej granicy plastyczności R_{eH} oraz granica wytrzymałości na rozciąganie f_u , która jest równa minimalnej wytrzymałości R_m . Norma PN-EN 1993-1-1 w tablicy 3.1 podaje nominalne wartości tych parametrów dla różnych gatunków stali walcowanych na gorąco. Parametry wytrzymałościowe stali walcowanej są uzależnione od grubości t ścianek elementów, wraz ze wzrostem grubości ścianki rośnie niejednorodność struktury i pogorszeniu ulega wytrzymałość elementu.
Podstawowe oznaczenie gatunku stali według europejskich norm hutniczych składa się z symbolu S oznaczającego, że mamy do czynienia ze stalą konstrukcyjną oraz liczby przedstawiającej granicę plastyczności stali w MPa. Przykładowo stal konstrukcyjna niestopowa według PN-EN 10025-2 o granicy plastyczności równej 355MPa jest oznaczana jako S355. W oznaczeniu stali mogą pojawić się również symbole dodatkowe, które oznaczają grupę jakościową stali oraz cechy gatunku stali. Norma PN-EN 1993-1-1 dotyczy stali o granicy plastyczności do 460 MPa. W realizowanym programie predefiniowano gatunki stali niestopowej według PN-EN 10025-2: S235, S275, S355, S450 zgodnie z Tablicą 3.1 wspomnianej już normy. Do wyznaczenia odkształceń przy uplastycznieniu użyto modułu sprężystości $E = 210\text{GPa}$, natomiast odkształcenia po osiągnięciu granicznej wytrzymałości określono zgodnie z podanymi wcześniej wymaganiami normowymi.



Rys. 12 Bilinearna zależność $\sigma - \epsilon$ dla stali konstrukcyjnej [1]

Do wyznaczania nośności przekroju poprzecznego, niezależnie od klasy norma PN-EN 1993-1-1 zaleca stosowanie częściowego współczynnika bezpieczeństwa $\gamma_{M0} = 1.00$.

2.5 Twierdzenie Greena o zamianie całki powierzchniowej na krzywoliniową

Warunki równoważności sił wewnętrznych i zewnętrznych wprowadzone we wzorach (10), (11), (12) zawierają całkowanie naprężeń po powierzchniach komponentów przekroju. Jedną z metod rozwiązania takich całek podwójnych dla przekrojów o dowolnej geometrii jest ich zamiana na całki krzywoliniowe po obwodzie poszczególnych komponentów. Zamianę całki powierzchniowej na całkę krzywoliniową można dokonać za pomocą twierdzenia Green'a. Mając już całkę liniową po obwodzie komponentu i zakładając, że jego geometria dopuszcza co najwyżej wielokąty możemy dokonać całkowania analitycznego lub stosując jedną z popularnych metod numerycznych np. metodę kwadratów, trapezów, etc. W realizowanym przez nas programie zastosowana będzie metoda numeryczna rozwiązywania całek liniowych Gaussa-Legendre. Twierdzenie Greena można wyrazić wzorem (48).

$$\oint_C P(x,y)dx+Q(x,y)dy=\iint_D\left(\frac{\delta Q}{\delta x}-\frac{\delta P}{\delta y}\right)dxdy \quad (48)$$

Gdzie C będzie pozytywnie zorientowaną, odcinkami gładką, prostą krzywą zamkniętą, a D będzie obszarem ograniczonym przez tą krzywą. Natomiast P i Q są funkcjami dwóch zmiennych (x,y) określonymi na otwartym obszarze zawierającym D i mają tam ciągłe pochodne cząstkowe. Twierdzenie Greena można zastosować do wzorów (10), (11), (12) poprzez przyjęcie odpowiednich wartości funkcji $P(x,y)$ oraz $Q(x,y)$. W celu otrzymania siły osiowej $N_{A,i}$ zamiast całkowania naprężeń $\sigma_x(y)$ po powierzchni i-tego komponentu A_i zamienia się całkę podwójną na całkę krzywoliniową stosując $P(y,z)=0$ oraz $Q(y,z)=y\cdot\sigma_x(z)$ w (48).

$$N_{A,i}=\oint_{Ci} y\cdot\sigma_x(z)\,dz=\iint_{A_i}\sigma_x(z)\,dydz \quad (49)$$

W analogiczny sposób można otrzymać momenty $M_y,-M_z$ podstawiając we wzorze (48) odpowiednio $P(y,z)=0,\ Q(y,z)=y\cdot z\cdot\sigma_x(z)$ oraz $P(y,z)=0,\ Q(y,z)=\frac{1}{2}\cdot y^2\cdot\sigma_x(z)$.

$$M_{y,A,i}=\oint_{Ci} y\cdot z\cdot\sigma_x(z)\,dz=\iint_{A_i} z\cdot\sigma_x(z)\,dydz \quad (50)$$

$$-M_{z,A,i}=\frac{1}{2}\oint_{Ci} y^2\cdot\sigma_x(z)\,dz=\iint_{A_i} y\cdot\sigma_x(z)\,dydz \quad (51)$$

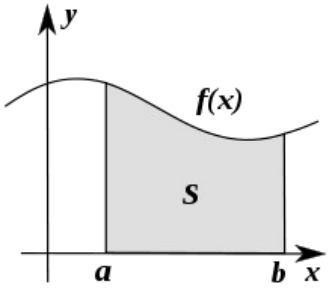
Przyjmując XCY jako centralny układ współrzędnych przekroju zespolonego zgodnie z pracą V.K. Papanikolaou'a [1] powyższe warunki równoważności dla i-tego komponentu powierzchniowego można by zapisać w sposób zagregowany jak w (52).

$$R_{A,i}=\frac{1}{r+1}\oint_{Ci} x^{r+1}\cdot y^s\cdot\sigma(y)\,dy=\iint_{A,i} x^r\cdot y^s\cdot\sigma(y)\,dxdy \quad (52)$$

Gdzie R jest wynikową siłą wewnętrzną zależną od wartości parametrów (r,s): N dla (0,0), Mx dla (0,1) oraz -My dla (1,0). W zależności (52) przyjęto następujące wartości funkcji dwóch zmiennych z twierdzenia Greena $P(x,y)=0$ oraz $Q(x,y)=\frac{1}{r+1}\cdot x^{r+1}\cdot y^s\cdot\sigma(y)$.

2.6 Metoda numeryczna Gaussa-Legendre rozwiązywania całek liniowych

Całkowanie numeryczne pozwala na przybliżone obliczanie całek oznaczonych, w przypadku całek pojedynczych nazywane jest również kwadraturą np. kwadratura Gauss-Legendre.



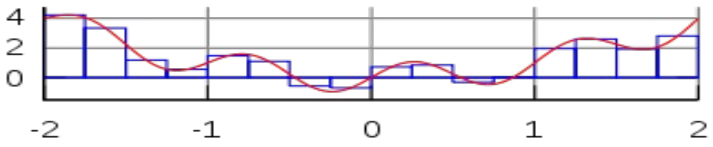
Rys. 13 Graficzna reprezentacja pojedynczej całki oznaczonej [12]

Poprzez rozwiązanie pojedynczej całki oznaczonej $\int_a^b f(x)\,dx$ rozumie się znalezienie pola powierzchni S pod wykresem funkcji $f(x)$ w przedziale $[a,b]$. W arytmetyce komputerowej nie ma możliwości zaimplementowania całkowania analitycznego, które pozwalałoby na otrzymywanie dokładnych wyników dla dowolnie zdefiniowanej funkcji podcałkowej. Stąd potrzeba stosowania metod numerycznych obliczania całek oznaczonych dających przybliżone wyniki z pewnym stopniem dokładności. Wśród podstawowych kwadratur można wyróżnić: metodę punktu środkowego (ang. mid-point rule) zwaną również metodą kwadratów (ang. rectangle rule) (Rys. 14), metodę trapezów (trapezoidal rule) (Rys.15), metodę Simpsona (ang. Simpson's rule) (Rys. 16).
W metodzie kwadratów funkcja interpolującą rzeczywistą funkcję podcałkową jest odcinkami stała. Przedział całki oznaczonej $[a,b]$ dzieli się na n podprzedziałów równej długości $d=(b-a)/n$. Dla każdego podprzedziału konstruujemy kwadrat aproksymujący powierzchnię pod funkcją podcałkową na tym podprzedziale w taki sposób, że funkcja $f(x)$ przecina lewy wierzchołek, środek (53) lub prawy wierzchołek górnej krawędzi kwadratu.

$$\int_a^b f(x)\,dx \approx \sum_{m=1}^{n} d\cdot f(x_m) \quad (53)$$

W powyższym wzorze x_m jest punktem środkowym każdego podprzedziału, a $d\cdot f(x_m)$ polem m-tego kwadratu.

$$x_m = a+(m-\frac{1}{2})\cdot d \quad (54)$$



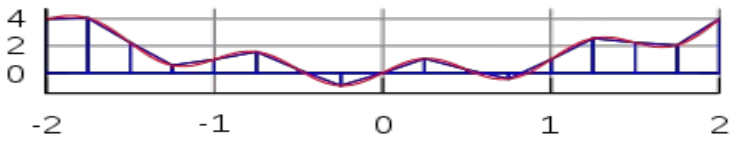
Rys. 14 Metoda kwadratów rozwiązywania całek oznaczonych [12]

Metoda trapezów umożliwia przybliżenie funkcji podcałkowej funkcją odcinkami liniową umożliwiając dyskretyzacje powierzchni S trapezami prostokątnymi. Można zastosować równomierną bądź nierównomierną siatkę trapezów. Przybliżoną wartość całki oznaczonej dla podziału równomiernego można obliczyć stosując wzór (55), gdzie $d=(b-a)/n$.

$$\int_a^b f(x)\,dx \approx \sum_{m=1}^{n} \frac{1}{2}\cdot d\cdot(f(x_m)+f(x_{m+1})) \quad (55)$$

W powyższym wzorze x_m jest współrzędną odpowiadającą lewej krawędzi m-tego trapezu (podprzedziału) (56), a wyrażenie pod sumą (55) jego polem powierzchni.

$$x_m = a+(m-1)\cdot d \quad (56)$$



Rys. 15 Metoda trapezów rozwiązywania całek oznaczonych [12]

Metoda Simpsona stosuje interpolacje funkcji podcałkowej $f(x)$ funkcją kwadratową tj. parabolą $P(x)$ posiadającą trzy punkty wspólne z $f(x)$: $(a, f(a))$, $(\frac{a+b}{2}, f(\frac{a+b}{2}))$, $(b, f(b))$. Przy podziale przedziału $[a, b]$ na parzystą liczbę n podprzedziałów całkowanie numeryczne metodą Simpsona można wyrazić wzorem (57), gdzie $d=(b-a)/n$.

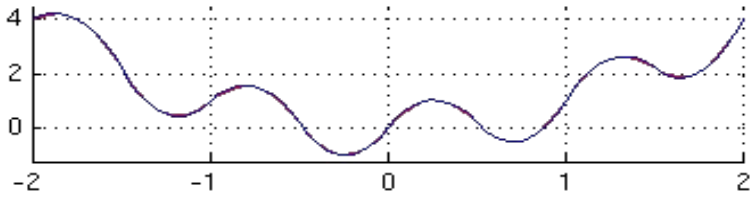
$$\int_a^b f(x)\,dx \approx \frac{d}{3}[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+\ldots+4f(x_{n-1})+f(x_n)] \quad (57)$$

W powyższym wzorze x_m dane jest wzorem (58).

$$x_m = a + m \cdot d \quad (58)$$

Przy zastosowaniu najmniejszej dopuszczalnej parzystej liczby przedziałów n = 2 powyższy formuła przyjmie postać daną wzorem (59).

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}[f(a)+4f(\frac{a+b}{2})+f(b)] \quad (59)$$



Rys. 16 Metoda Simpsona rozwiązywania całek oznaczonych [12]

Kwadratura Gaussa stanowi kolejną odmianę całkowania numerycznego, która dopuszcza nierównomierny rozkład punktów interpolacji funkcji podcałkowej. Daje ona dokładne wyniki dla wielomianów stopnia co najwyżej 2n-1 przy zastosowaniu n punktów Gaussa. W ogólności kwadratura pozwala na obliczenie całki oznaczonej jako sumy ważonej wartości funkcji podcałkowej w zadanych punktach. Na przedziale [-1,1] kwadratura Gaussa opisana jest wzorem (60), który można uogólnić na dowolny przedział $[a, b]$ stosując równość (61) i w wyniku otrzymując zależność daną wzorem (62), gdzie x_m dane jest wzorem (63).

$$\int_{-1}^{1} f(\xi)\,d\xi \approx \sum_{m=1}^{nG} w_m \cdot f(\xi_m) \quad (60)$$

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f(\frac{b+a}{2}+\frac{b-a}{2}\cdot\xi)\,d\xi \quad (61)$$

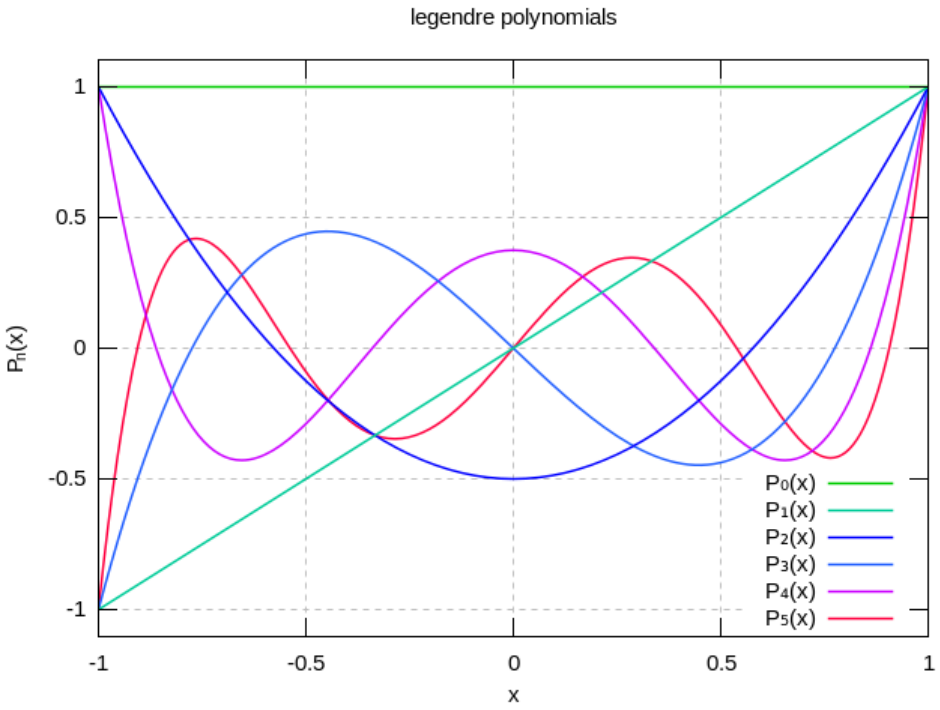
$$\int_a^b f(x)\,dx \approx \frac{1}{2}\cdot(b-a)\cdot\sum_{m=1}^{nG} w_m \cdot f(x_m) \quad (62)$$

$$x_m = \frac{1}{2}\cdot(b+a)+\frac{\xi_m}{2}\cdot(b-a) \quad (63)$$

Do obliczania całek oznaczonych metodą Gaussa najczęściej wykorzystuje się wielomiany Legendre, które jak już wspomniano dla funkcji podcałkowych będących również wielomianami

stopnia co najwyżej 2n-1 (gdzie n to liczba punktów Gaussa) dają dokładne wyniki. Oznaczając wielomiany Legendre jako $P_n(\xi)$, n oznacza stopień wielomianu. Wykresy wielomianów Legendre do stopnia 5 włącznie przedstawiono na poniższym wykresie (Rys. 17). Warto zauważyć iż wielomiany te są normalizowane w taki sposób aby wszystkie przechodziły przez punkt (1,1). Poszczególnym wielomianom odpowiada $nG=n$ punktów Gaussa ξ_m , będących kolejnymi pierwiastkami danego wielomianu $(\xi_m, 0)$. Każdemu punktowi Gaussa przyporządkowana jest waga w_m pojawiająca się w zależnościach (60), (62) i dana poniższym równaniem (64).

$$w_m = \frac{2}{(1-\xi_m^2)\cdot(P_n'(\xi_m))^2} \quad (64)$$



Rys. 17 Wykresy pierwszych sześciu wielomianów Legendre [15]

W poniższej tablicy zestawiono wartości współrzędnych ξ_m pierwiastków i odpowiadające im wagi w_m dla wielomianów Legendre do stopnia 5 włącznie. Tablica 2 zawiera wystarczającą liczbę informacji do zaimplementowania realizowanego programu w sposób pozwalający na uzyskanie dokładnych wyników całkowania jako, że stosowane funkcje podcałkowe przy założeniu normowych zależności naprężenie-odkształcenie będą wymagać co najwyżej trzech punktów Gausaa nG = 3.

Tablica 2. Punkty Gaussa i wagi odpowiadające wielomianą Legendre $P_n(\xi)$ do stopnia n = 5 włącznie

Wielomian Legendre	Liczba punktów Gauss'a [nG]	Współrzędne punktów $[\xi_m]$	Wagi $[w_m]$
$P_0(\xi)$	0	-	-

$P_1(\xi)$	1	0	2
$P_2(\xi)$	2	$-\sqrt{\frac{1}{3}},\ +\sqrt{\frac{1}{3}}$	1
$P_3(\xi)$	3	0	$\frac{8}{9}$
		$-\sqrt{\frac{3}{5}},\ +\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
$P_4(\xi)$	4	$-\sqrt{\frac{(3-2\sqrt{\frac{6}{5}})}{7}},\ +\sqrt{\frac{(3-2\sqrt{\frac{6}{5}})}{7}}$	$\frac{18+\sqrt{30}}{36}$
		$-\sqrt{\frac{(3+2\sqrt{\frac{6}{5}})}{7}},\ +\sqrt{\frac{(3+2\sqrt{\frac{6}{5}})}{7}}$	$\frac{18-\sqrt{30}}{36}$
$P_5(\xi)$	5	0	$\frac{128}{225}$
		$-\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}},\ +\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$	$\frac{322+13\sqrt{70}}{900}$
		$-\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}},\ +\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	$\frac{322-13\sqrt{70}}{900}$

3. NUMERYCZNY ZAPIS PRZEKROJU O DOWOLNEJ ZŁOŻONOŚCI

Realizowany program powinien pozwalać na zdefiniowanie jak najbardziej ogólnego przekroju zespolonego, jednocześnie warto ograniczyć złożoność algorytmu poprzez kierowanie się jego przeznaczeniem i uwzględnienie rzeczywistych geometrii typowych konstrukcji zespolonych. Zgodnie z propozycją V.K. Papanikolaou'a [1] w programie zdefiniowano trzy rodzaje komponentów: powierzchnie (ang. surfaces), elementy liniowe (ang. lines), zbrojenie (ang. fiber groups). Jako, że przekrój może zawierać dowolną liczbę każdego z tych komponentów utworzono dla nich odpowiednio trzy typy: Surface, Line, FiberGroup. Wszystkie one posiadają cechy wspólne efektem czego dziedziczą publicznie po klasie nadrzędnej Points. (Rys. 18)

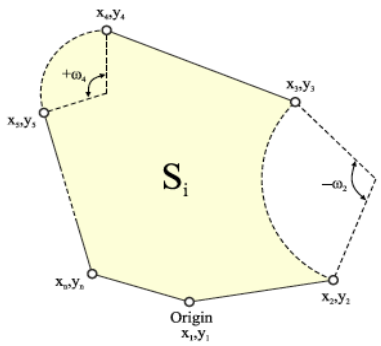
Points
points : QVarLengthArray<Point *>
+ addPoint(double x, double y) : void + getPointsArray(void) : Point ** + numberOfPoints(void) : int

Rys. 18 Diagram UML klasy nadrzędnej komponentów przekroju

Obiekt typu Points przechowuje dowolną liczbę punktów (wskaźniki na obiekty Point) określających komponent przekroju wewnątrz kontenera typu QVarLengthArray<T>. Klasa posiada zdefiniowane metody obiektowe umożliwiające dodanie nowych wierzchołków, pobieranie tablicy wskaźników na obiekty wierzchołków oraz liczbę aktualnie przechowywanych wierzchołków.

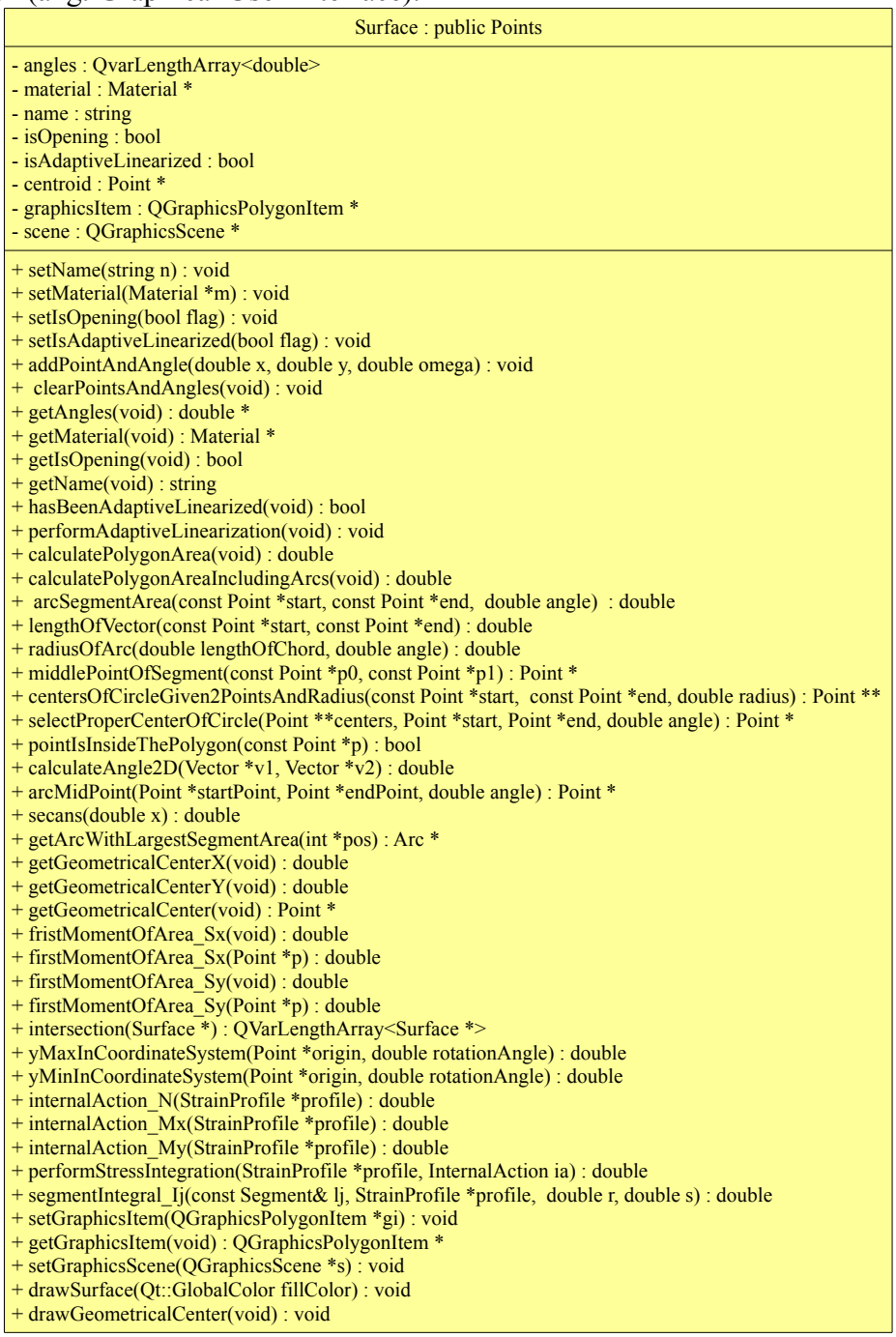
3.1 Komponenty powierzchniowe

Komponenty powierzchniowe $(S_1, S_2, \ldots S_n)$ to elementy przekroju o znacznej powierzchni tj. beton, stal konstrukcyjna np. dwuteowniki stalowe. Realizowany program dopuszcza stosowanie komponentów powierzchniowych w postaci wielokątów z dowolną liczbą wierzchołków, których krawędzie mogą przyjmować postać łuków wklęsłych lub wypukłych (Rys. 19). Definiowanie i-tej powierzchni przekroju S_i odbywa się w postaci ciągu elementów (x_j, y_j, ω_i) gdzie x_j, y_j są współrzędnymi j-tego wierzchołka, a ω_j jest kątem środkowym łuku występującego po tym wierzchołku. Kąt ten przyjmuje wartości dodatnie $(0, \pi]$ dla łuków wypukłych tj. zwiększających powierzchnię wielokąta, oraz wartości ujemne $[-\pi, 0)$ dla łuków wklęsłych tj. zmniejszających powierzchnię wielokąta. W przypadku odcinków prostych wartość ω_j jest pomijana poprzez przypisanie jej wartości równej 0. Poszczególne wierzchołki i kąty komponentu są podawane w kolejności przeciwnej do ruchu wskazówek zegara (ang. CCW) $S_i = [(x_1, y_1, \omega_1), (x_2, y_2, \omega_2), \ldots, (x_n, y_n, \omega_n)]$.



Rys. 19 Sposób definiowania komponentu powierzchniowego (ang. surface) [1]

W kodzie powyższe wierzchołki (obiekty typu Point) i kąty (typu double) w radianach zdefiniowano niezależnie w osobnych kontenerach QVarLengthArray<T>. Tablicę wierzchołków we wspomnianej już klasie nadrzędnej Points, natomiast tablicę kątów w dziedziczącej po niej klasie Surface w taki sposób, że odpowiadające sobie punkty i kąty znajdują się na odpowiadających sobie pozycjach. Do obiektu Surface przypisany jest poprzez zmienną składową obiekt typu Material zawierający charakterystyki materiałowe. Komponenty powierzchniowe mogą służyć nie tylko do definiowania poszczególnych materiałów w przekroju zespolonym, ale również otworów (ang. voids). Stosuje się w tym celu zmienną flagową *isOpenning* typu bool, która jeżeli jest ustawiona na *true* to funkcje obliczające wartości sił przekrojowych zwracają 0. Typ Surface oprócz struktur danych przechowujących informacje o geometrii danej powierzchni posiada również liczne metody, które można pogrupować w następujący sposób: metody akcesorowe, metody wykonujące linearyzacje adaptacyjną komponentu, metody wyznaczające charakterystyki geometryczne przekroju, metody całkujące naprężenie po powierzchni elementu, metody rysujące przekrój w GUI (ang. Graphical User Interface).

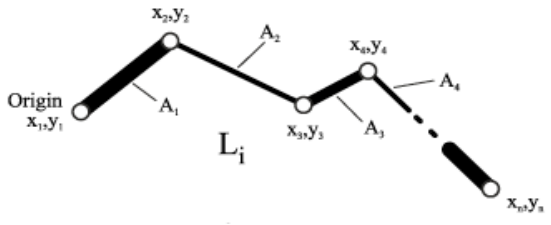


Rys. 20 Diagram UML klasy Surface (komponent powierzchniowy)

Reprezentacja klasy Surface przedstawiona jest w postaci diagramu UML na Rys. 20. Ważniejsze procedury zostaną omówione w dalszej części tej pracy.

3.2 Komponenty Liniowe

W realizowanym programie istnieje możliwość zdefiniowania wieloodcinkowego elementu liniowego, który ma zastosowanie np. podczas definiowania zbrojenia (uproszczenie wprowadzania danych biorąc pod uwagę, że rozstaw zbrojenia z reguły wynika i tak z wymagań normowych, a nie z nośności), taśm wzmacnianych włóknami (ang. FRP – fiber-reinforced polymer) w tym taśmy z włókien węglowych (ang. CFRP - carbon fiber-reinforced polymer). Komponenty liniowe reprezentowane są przez ciąg współrzędnych wierzchołków oraz powierzchni przypisanych odcinkowi występującemu za bieżącym wierzchołkiem $L_i=[(x_1, y_1, A_1), \ldots (x_n, y_n, A_n)]$ przy czym powierzchnia A_n odpowiadająca punktowi końcowemu elementu jest równa pomijana i równa 0.



Rys. 21 Sposób definiowania komponentu liniowego (ang. line) [1]

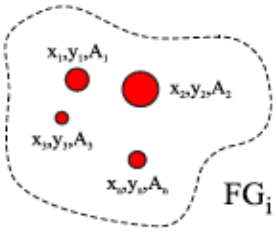
W kodzie typ Line dziedziczy podobnie jak pozostałe typy komponentów przekroju po wspólnej klasie nadrzędnej Points uzupełniając ją o tablice przechowującą powierzchnie A_i przypisane poszczególnym odcinkom. Do obiektów Line przypisany jest poprzez zmienna składową obiekt typu Material zawierający definicję materiału. Klasa Line oprócz struktur danych przechowujących informacje o geometrii komponentu posiada następujące rodzaje metod: metody akcesorowe, metody do wyznaczania charakterystyk geometrycznych, metody do całkowania naprężeń po powierzchni elementu. (Rys. 22)

Line : public Points
- areas : QvarLengthArray<double> - material : Material * - name : string
+ addPointAndArea(double x, double y, double area) : void + getAreas(void) : double * + setMaterial(Material *m) : void + getMaterial(void) : Material * + setName(string n) : void + getName(void) : string + clearPointsAndAreas(void) : void + getOrigin(void) : Point * + yMaxInCoordinateSystem(Point *origin, double rotationAngle) : double + yMinInCoordinateSystem(Point *origin, double rotationAngle) : double + internalAction_N(StrainProfile *profile) : double + internalAction_Mx(StrainProfile *profile) : double + internalAction_My(StrainProfile *profile) : double + performStressIntegration(StrainProfile *profile, InternalAction ia) : double + segmentIntegral_Ij(const Segment& lj, StrainProfile *profile, double r, double s, int lineIdx) : double + getTotalArea(void) : double + getGeometricalCenterX(void) : double + getGeometricalCenterY(void) : double + getGeometricalCenter(void) : Point * + firstMomentOfArea_Sx(void) : double + firstMomentOfArea_Sx(Point *p) : double + firstMomentOfArea_Sy(void) : double + firstMomentOfArea_Sy(Point *p) : double + getComponentCenterX(int i) : double + getComponentCenterY(int i) : double + getComponentCenter(int I) : Point *

Rys. 22 Diagram UML klasy Line (komponent liniowy)

3.3 Komponenty reprezentujące zbrojenia

Ostatnim typem komponentu, który może występować w przekroju zespolonym w realizowanym programie jest zbrojenie (ang. fiber groups). Pozwala on na definiowanie prętów zbrojeniowych jako zbioru punktów o współrzędnych (x_i, y_i) z przyporządkowaną im powierzchnią zbrojenia A_i. Zbiór ten można wyrazić abstrakcyjnie w następujący sposób (Rys. 23) $FG_i = [(x_1, y_1, A_1), \ldots, (x_n, y_n, A_n)]$. W kodzie podobnie jak w przypadku typów Surface i Line współrzędne punktów przechowywane są w osobnej tablicy QVarLengthArray<Point *> będącej składową klasy nadrzędnej Points jako wskaźniki na obiekty Point. W klasie FiberGroup zdefiniowana jest natomiast niezależna tablica z powierzchniami przekrojów prętów A_i. Elementy obu tablic powiązane są tymi samymi wartościami indeksów.



Rys. 23 Sposób definiowania zbrojenia (ang. fiber group) [1]

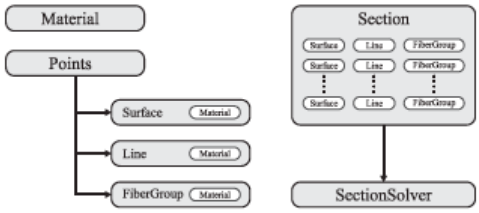
Do obiektów FiberGroup poprzez zmienną składową przypisywany jest obiekt typu Material określający gatunek stali zbrojeniowej i jej charakterystyki materiałowe. W klasie FiberGroup oprócz struktur danych służących do definiowania komponentów reprezentujących zbrojenie znajdują się również procedury, które można pogrupować w następujący sposób: metody akcesorowe, metody do wyznaczania charakterystyk geometrycznych, metody do całkowania naprężeń po powierzchniach przekrojów prętów zbrojeniowych. (Rys. 24)

FiberGroup : public Points
- areas : QVarLengthArray<double> - material : Material * - name : string
+ addPointAndArea(double x, double y, double area) : void + getAreas(void) : double * + setMaterial(Material *m) : void + getMaterial(void) : Material * + setName(string n) : void + getName(void) : string + clearPointsAndAreas(void) : void + getOrigin(void) : Point * + yMaxInCoordinateSystem(Point *origin, double rotationAngle) : double + yMinInCoordinateSystem(Point *origin, double rotationAngle) : double + internalAction_N(StrainProfile *profile) : double + internalAction_Mx(StrainProfile *profile) : double + internalAction_My(StrainProfile *profile) : double + performStressIntegration(StrainProfile *profile, InternalAction ia) : double + getTotalArea(void) : double + getGeometricalCenterX(void) : double + getGeometricalCenterY(void) : double + getGeometricalCenter(void) : Point * + firstMomentOfArea_Sx(void) : double + firstMomentOfArea_Sx(Point *p) : double + firstMomentOfArea_Sy(void) : double + firstMomentOfArea_Sy(Point *p) : double + getXCenterOfFiber(int i) : double + getYCenterOfFiber(int i) : double + getCenterOfFiber(int i) : Point *

Rys. 24 Diagram UML klasy FiberGroup (komponent zbrojenia)

3.4 Agregacja komponentów przekroju w obiekcie klasy Section

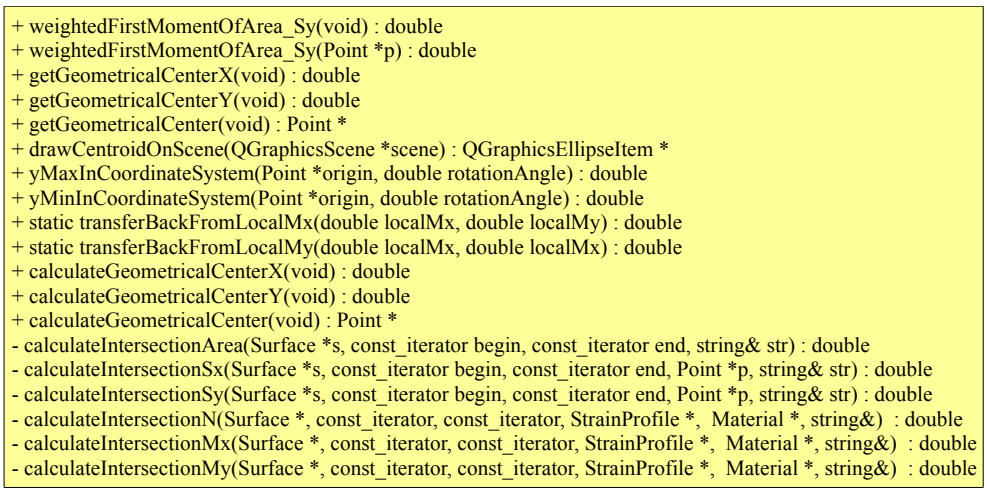
Przekrój zespolony może posiadać dowolną liczbę komponentów tj. obiektów klasy Surface, Line oraz FiberGroup. Wszystkie one dziedziczą po wspólnej klasie nadrzędnej Points oraz posiadają przypisany materiał tj. obiekt typu Material z jakiego są wykonane poprzez zmienną składową. Obiekt klasy Section stanowi faktyczną definicję całego przekroju agregując w swoich tablicach QvarLengthArray<Surface *>, QvarLengthArray<Line *>, QvarLengthArray<FiberGroup *> poszczególne typy komponentów. Klasa Section oprócz tego, że zawiera struktury danych umożliwiające zgrupowanie całego przekroju zespolonego w jednym obiekcie pozwala również na wykonywanie operacji tj. wyznaczanie charakterystyk geometrycznych czy sił wewnętrznych w przekroju drogą całkowania naprężeń po powierzchni przekroju. Poszczególne metody podczas wykonywania obliczeń uwzględniają wyniki zwracane przez odpowiadające metody obiektów komponentów, np. podczas wyznaczania momentu statycznego całego przekroju funkcja klasy Section w odpowiedniej pętli sumuje momenty statyczne od wszystkich elementów Surface, Line, FiberGroup przechowywanych w wewnętrznych tablicach obiektu Section.



Rys. 25 Obiektowy zapis przekroju zespolonego [1]

Po klasie Section dziedziczy dodatkowo klasa SectionSolver, która opakowuje procedury związane z poszukiwaniem rozwiązania dla zadanego w klasie nadrzędnej przekroju zespolonego. Operacje udostępniane przez obiekt tej klasy pozwalają na poszukiwanie dopuszczalnych granicznych odkształceń rozciągających i ściskających dla całego przekroju w zależności od położenia środka ciężkości i obrotu lokalnego układu współrzędnych oraz w dalszej kolejności na konstruowanie granicznych profili odkształceń (obiekty klasy StrainProfile) na ich podstawie. Ponadto znajdują się tam metody umożliwiające zastosowanie całkowania do tak wyznaczonych profili odkształceń za pośrednictwem metod całkujących w klasie nadrzędnej Section. Klasa SectionSolver posiada również funkcje umożliwiające uporządkowanie wyników w taki sposób by mogły zostać wykorzystane do konstruowania powierzchni i krzywych interakcji.

Section
surfaces: QvarLengthArray<Surface *> # lines : QvarLengthArray<Line *> # fiberGroups : QvarLengthArray<FiberGroup *> - centroid : Point *
+ addSurface(Surface *) : void + addLine(Line *) : void + addFiberGroup(FiberGroup *) : void + getSurfaces(void) : Surface ** + getLines(void) : Line ** + getFiberGroups(void) : FiberGroup ** + getNumberOfSurfaces(void) : int + getNumberOfLines(void) : int + getNumberOfFiberGroups(void) : int + defineCentroid(void) : Point * + summateInternalActions_N(StrainProfile *profile) : double + summateInternalActions_Mx(StrainProfile *profile) : double + summateInternalActions_My(StrainProfile *profile) : double + weightedArea(void) : double + weightedFirstMomentOfArea_Sx(void) : double + weightedFirstMomentOfArea_Sx(Point *p) : double



Rys. 26 Diagram UML klasy Section agregującej komponenty przekroju.

Ważniejsze procedury klas Section (Rys. 26) oraz SectionSolver zostaną omówone w daleszej części pracy.

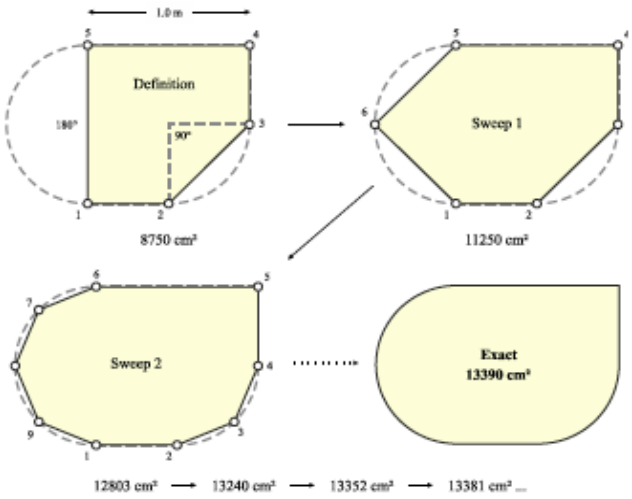
3.5 Linearyzacja komponentów powierzchniowych przekroju

Brzeg komponentu powierzchniowego może oprócz odcinków prostych składać się również z łuków wklęsłych i wypukłych. Łuki te należy aproksymować odcinkami liniowymi by zachować zgodność ze stosowanym na późniejszym etapie algorytmem całkowania numerycznego Gaussa-Legendre oraz twierdzeniem Greena. W tym celu dla każdego komponentu powierzchniowego wykonywana jest procedura linearyzacji adaptacyjnej. W ogólności proponowana metoda polega na podziale istniejących łuków na dwie równe części poprzez wstawienie dodatkowych punktów środkowych dla kolejnych łuków. (Rys. 27) Podziały w poszczególnych krokach dokonywane są dla łuków posiadających największą powierzchnię segmentu (ang. arc segment) aż do momentu gdy powierzchnia otrzymanego w wyniku podziału wielokąta będzie różniła się od rzeczywistej powierzchni (uwzględniającej łuki) jedynie o żądaną wartość błędu (65). Wprowadzona została tutaj pewna modyfikacja w stosunku do algorytmu V.K. Papanikolaou'a, który jako warunek zbieżności procesu linearyzacji proponował porównywanie powierzchni wielokątów otrzymywanych w kolejnych iteracjach aż do osiągnięcia założonej tolerancji błędu (66).

$$\left|\frac{A_{total}-A^{i}_{polygon}}{A_{total}}\right|<A_{TOL} \quad (65)$$

$$\left|\frac{A^{i+1}_{polygon}-A^{i}_{polygon}}{A^{i}_{polygon}}\right|<A_{TOL} \quad (66)$$

Przyjęty w realizowanym programie warunek zbieżności to $A_{TOL}=0.001$. Stosowany algorytm linearyzacji do wyszukiwania kolejnych łuków na których dokonywany ma być podział używa funkcji *Arc *getArcWithLargestSegmentArea(int *pos)*. Następnie dla wybranego łuku oblicza położenie jego punktu środkowego (ang. arc midpoint) za pomocą metody *Point *arcMidPoint(Point *start, Point *end, double angle)* i wstawia go do tablicy wierzchołków komponentu powierzchniowego pomiędzy punktami początku (*pos*) i końca łuku (*pos+1*). Dodatkowo powstałym w wyniku podziału odcinkom przypisywane są nowe wartości kątów równe *angle/2*. Powierzchnia segmentu łuku tj. obszaru ograniczonego przez łuk oraz prostą przechodzącą przez jego końce dana jest wzorem (67). Gdzie promień *r* łuku wyznaczany jest na podstawie długości cięciwy (ang. chord) i kąta środkowego ze wzoru (68). *W* to długość cięciwy łuku.



Rys. 27 Metoda linearyzacji komponentu powierzchniowego [1]

$$A_{sector}=\frac{1}{2}\cdot r^2\cdot(\theta-\sin(\theta)) \quad (67)$$

$$r=\frac{W}{(2\cdot\sin(\frac{\theta}{2}))} \quad (68)$$

Punkt środkowy łuku można wyznaczyć używając iloczynu skalarnego. Współrzędne tego punktu dane są zależnością (69). Gdzie $C=(c_x, c_y)$ to środek okręgu, $A=(2a_x, 2a_y)$ początek łuku, $B=(2b_x, 2b_y)$ koniec łuku, $M=(m_x, m_y)$ to szukany punkt środkowy, a $\theta=2\alpha$.

$$m=c+(a+b-c)\cdot sec(\alpha) \quad (69)$$

Do wyznaczania środka okręgu $C=(c_x, c_y)$ wystarcza znajomość dowolnych dwóch punktów leżących na tym okręgu (w tym przypadku początek *A* i koniec *B* łuku) oraz długość promienia *r* (68). Długość cięciwy we wzorze na *r* to po prostu długość wektora łączącego punkty *A* i *B*. Należy znaleźć środek (x_m, y_m) odcinka $\bar{AB}$, wyznaczyć wektor $\vec{AB}$ oraz wektor do niego prostopadły $\vec{v}$ (70) i poddać go normalizacji (71).

$$\vec{AB}=[x_1-x_{0,}\, y_1-y_0]--->\vec{v}=[y_1-y_{0,}\, x_0-x_1] \quad (70)$$

$$\vec{v_{norm}}=\frac{\vec{v}}{||\vec{v}||}=\frac{\vec{v}}{\sqrt{v_x^2+v_y^2}} \quad (71).$$

We wzorze (70) można przyjąć $x_0=2a_x, y_0=2a_y, x_1=2b_x, y_1=2b_y$. Należy zauważyć, że poza przypadkiem kiedy $C=(x_m, y_m)$ istnieją dwa środki okręgów $C_{1,}C_2$. Obydwa są położone na prostej, której wektorem kierunkowym jest $\vec{v_{norm}}$ w odległości *d* (72) od środka odcinka (x_m, y_m) .

$$d=\sqrt{r^2-(\frac{W}{2})^2} \quad (72)$$

Ostatecznie środki okręgu $C_{1,}C_2$ dane są wzorami (73), (74).

$$C_1=(x_m, y_m)+d\cdot\vec{v_{norm}} \quad (73)$$

$$C_2=(x_m, y_m)-d\cdot\vec{v_{norm}} \quad (74)$$

Przed wyznaczeniem punktu środkowego łuku należy wybrać poprawny środek okręgu C spośród dwóch możliwych opcji C_1, C_2 na podstawie kąta środkowego θ oraz współrzędnych końców A i B łuku. Co do zasady przy zapisie wierzchołków przeciwnie do ruchu wskazówek zegara i założeniu że ujemna wartość kąta oznacza odejmowanie powierzchni, a dodatnia wartość jej powiększanie punkt C powinien leżeć po lewej stronie odcinka $\bar{AB}$ dla $\theta<0$, a po prawej dla $\theta>0$. W obu przypadkach może on również wypadać w środku odcinka np. $\theta=180^\circ$.

4. NUMERYCZNE WYZNACZANIE CHARAKTERYSTYK GEOMETRYCZNYCH

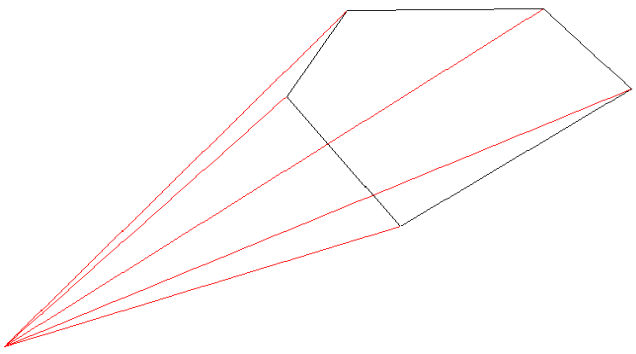
Do wyznaczenia charakterystyk geometrycznych takich jak pole powierzchni A, momenty statyczne S_x, S_y , momenty bezwładności I_x, I_y, I_{xy} , środek geometryczny C poszczególnych komponentów przekroju jak i ich ważonych odpowiedników dla całego przekroju zespolonego potrzebne są ogólniejsze wzory niż te stosowane zazwyczaj. Formuły te powinny umożliwiać wyznaczenie charakterystyk dla dowolnych wielokątów (ang. polygon).

4.1 Charakterystyki geometryczne komponentów powierzchniowych

Komponenty powierzchniowe reprezentowane są w postaci wielokątów. Ewentualne odcinki łukowe zostają poddane procesowi linearyzacji zgodnie z pkt. 3.5. Pole powierzchni prostego (ang. non self-intersecting) wielokąta o n wierzchołkach można wyznaczyć stosując zależność (75). W celu dodatkowego uwzględnienia segmentów łukowych np. wyznaczając całkowite pole powierzchni A_{total} można skorzystać ze wzoru (67).

$$A_{polygon}=\frac{1}{2}\cdot\sum_{i=0}^{n-1}(x_i\cdot y_{i+1}-x_{i+1}\cdot y_i) \quad (75)$$

Wzór (75) został zaimplementowany w kodzie przez funkcję *double calculatePolygonArea(void)*. Momenty statyczne S_x, S_y wielokąta o n wierzchołkach można wyznaczyć stosując wzory (76), (77) jako suma momentów statycznych trójkątów utworzonych przez wektory wodzące kolejnych wierzchołków wielokąta i odpowiadających im boków wielokąta (Rys. 28).



Rys. 28 Zasada obliczania momentów statycznych i bezwładności wielokąta

$$S_x = \frac{1}{6} \cdot \sum_{i=0}^{n-1} [(y_i + y_{i+1}) \cdot (x_i \cdot y_{i+1} - x_{i+1} \cdot y_i)] \quad (76)$$

$$S_y = \frac{1}{6} \cdot \sum_{i=0}^{n-1} [(x_i + x_{i+1}) \cdot (x_i \cdot y_{i+1} - x_{i+1} \cdot y_i)] \quad (77)$$

Warto zauważyć, że po usunięciu w powyższych wzorach składników $\frac{1}{3} \cdot (y_i + y_i + 1)$, $\frac{1}{3} \cdot (x_i + x_i + 1)$ odpowiadających odległościom środka ciężkości i-tego trójkąta od przyjętego układu współrzędnych otrzymujemy wzór (75) wyrażający pole powierzchni wielokąta jako sumę wspomnianych trójkątów. W analogiczny sposób wyznacza się momenty bezwładności I_x, I_y, I_{xy} (78), (79), (80).

$$I_x = \frac{1}{12} \sum_{i=0}^{n-1} [(x_{i+1} - x_i) \cdot (y_{i+1} + y_i) \cdot (y_{i+1}^2 + y_i^2)] \quad (78)$$

$$I_y = \frac{1}{12} \sum_{i=0}^{n-1} [(y_{i+1} - y_i) \cdot (x_{i+1} + x_i) \cdot (x_{i+1}^2 + x_i^2)] \quad (79)$$

$$I_{xy} = \frac{1}{24} \sum_{i=0}^{n-1} [(x_i - x_{i+1}) \cdot (3x_{i+1} y_{i+1}^2 + x_i y_{i+1}^2 + 2x_{i+1} y_i y_{i+1} + 2x_i y_i y_{i+1} + x_{i+1} y_i^2 + 3x_i^2 y_i^2)] \quad (80)$$

Środek ciężkości i-tego komponentu $C_{S,i}$ można łatwo wyliczyć dzieląc momenty statyczne ze wzorów (76), (77) przez pole powierzchni obliczone według (75).

$$C_{Si} = (x_{c,Si} = \frac{S_y}{A_{polygon}}, y_{c,Si} = \frac{S_x}{A_{polygon}}) \quad (81)$$

W kodzie powyższe wzoru zostały zaimplementowane w klasie Surface za pośrednictwem funkcji: `double getGeometricalCenterX(void)`, `double getGeometricalCenterY(void)`, `double firstMomentOfArea_Sx(Point *p)`, `double firstMomentOfArea_Sy(Point *p)`.
Warto zauważyć, że wymienione metody umożliwiają wyznaczanie momentów statycznych względem dowolnie wskazanego punktu `Point *p`.

4.2 Charakterystyki geometryczne komponentów liniowych

W przypadku komponentów liniowych sytuacja jest zdecydowanie prostsza. Pole powierzchni całego komponentu można wyznaczyć jako suma pól powierzchni odpowiadających poszczególnym odcinkom (82).

$$A_{line} = \sum_{i=0}^{n-1} A_i \quad (82)$$

Natomiast momenty statyczne S_x, S_y jako suma iloczynów pola powierzchni i-tego odcinka i odległości od jego środka do odpowiedniej osi przyjętego układu współrzędnych (83).

$$S_x = \sum_{i=0}^{n-1} A_i (y_{i,c} - y_o) \quad S_y = \sum_{i=0}^{n-1} A_i (x_{i,c} - x_o) \quad (83)$$

Współrzędne $(x_{i,c}, y_{i,c})$ są środkiem i-tego odcinka, a (x_o, y_o) początkiem przyjętego układu odniesienia. Powyższe zależności zaimplementowano w klasie Line w funkcjach: `double getTotalArea(void)`, `double firstMomentOfArea_Sx(Point *p)`, `double firstMomentOfArea_Sx(Point *p)`. Środek ciężkości (84) całego komponentu liniowego wyznaczamy dzieląc otrzymane momenty statyczne (83) przez całkowite pole powierzchni (82).

$$C_{Li} = (x_{c,Li} = \frac{S_y}{A_{line}}, y_{c,Li} = \frac{S_x}{A_{line}}) \quad (84)$$

4.3 Charakterystyki geometryczne zbrojenia

Równie prosto można wyznaczyć charakterystyki geometryczne dla komponentów definiujących zbrojenie. Całkowita powierzchnia zbrojenia dla danego komponentu to A_{fiber} dana wzorem (85).

$$A_{fiber} = \sum_{i=0}^{n-1} A_i \quad (85)$$

Jest to więc suma powierzchni przekroju poszczególnych prętów zbrojeniowych wchodzących w skład komponentu. Momenty statyczne S_x, S_y wyznacza się analogicznie jak w przypadku elementów liniowych z tym, że środki ciężkości i-tego pręta zbrojeniowego dane są explicite w strukturze danych reprezentującej komponent zbrojenia (86).

$$S_x = \sum_{i=0}^{n-1} A_i (y_i - y_o) \quad S_y = \sum_{i=0}^{n-1} A_i (x_i - x_o) \quad (86)$$

W końcu geometryczny środek ciężkości przekroju można wyznaczyć ze wzoru (87).

$$C_{FGi} = (x_{c,FGi} = \frac{S_y}{A_{fiber}}, y_{c,FGi} = \frac{S_x}{A_{fiber}}) \quad (87)$$

W kodzie powyższe zależności zaimplementowano w klasie FiberGroup w funkcjach o tych samych nazwach jak w klasie Line (podanych w podpunkcie 4.3).

4.4 Ważone charakterystyki geometryczne całego przekroju zespolonego

W przekrojach zespolonych jak już wspominano w rozdziale 2 niniejszej pracy wyznacza się tzw. ważone charakterystyki geometryczne $A^*, S_x^*, S_y^*, C^*, I_x^*, I_y^*, I_{xy}^*$. Z punktu widzenia realizowanego programu istotne będą jedynie pierwsze cztery wielkości. W celu wyznaczenia charakterystyk dla całego przekroju zespolonego należy uwzględnić wpływ od poszczególnych komponentów: powierzchniowych, liniowych i zbrojenia. Każdemu komponentowi przypisana jest odpowiednia waga wyrażona jako iloraz moduł sprężystości i-tego materiału oraz materiału odniesienia $n = E_i / E$. Gdzie materiałem odniesienia jest ten przypisany do pierwszego komponentu powierzchniowego S_0. Algorytm postępowania jest analogiczny dla przypadku wszystkich trzech pierwszych charakterystyk stąd zostanie on zaprezentowany na przykładzie zmiennej X^*.

1) sumowanie po wszystkich komponentach powierzchniowych, gdzie $X_{S,i}$ to charakterystyka i-tego komponentu, a $n_{S,i}$ odpowiadająca mu waga. Jeżeli komponent powierzchniowy jest otworem jego waga przyjmuje wartość równą 0, stąd jego wpływ nie będzie uwzględniany, natomiast wpływ od materiałów znajdujących się pod otworem zostanie uwzględniony po zastosowaniu pkt. 2

$$X^* = \sum_{i=0}^{nS} n_{S,i} \cdot X_{S,i} \quad (88)$$

2) następnie należy odjąć/dodać wpływy od przecięć komponentów powierzchniowych. W tym celu stosowany jest generator wszystkich możliwych kombinacji takich przecięć (opisany dokładniej w pkt. 4.5) Każdemu przecięciu przypisany jest materiał który definiuje odpowiadającą mu wagę. Przecięcie wielokątów jest również wielokątem. Wynikiem przecięcia dla pojedynczej kombinacji może być liczba wielokątów ≥ 0 .

$$X^* \mathrel{+}= \sum_{i=0}^{nIntersecions} \text{-/+} n_{Inter,i} \cdot X_{Inter,i} \quad (89)$$

3) dodanie wpływów od elementów liniowych i zbrojenia na końcową wartość charakterystyki ważonej.

$$X^* \mathrel{+}= \sum_{i=0}^{nL} n_{L,i} \cdot X_{L,i} \quad (90)$$

$$X^* \mathrel{+}= \sum_{i=0}^{nFG} n_{FG,i} \cdot X_{FG,i} \quad (91)$$

4) dodając wpływy od poszczególnych elementów liniowych i zbrojenia odjąć wpływ z powierzchni o odpowiadającej wielkości przy założeniu materiału jak dla komponentu w którym te elementy są osadzone. Przyjęto założenie, że elementy liniowe i zbrojenie mogą być zdefiniowane jedynie jako osadzone w innym jednorodnym komponencie powierzchniowym. Wyszukiwanie tej powierzchni wykonywać należy w kierunku odwrotnym do kolejności ich zdefiniowania w tablicowej strukturze danych (powierzchnie są tam umieszczone w kolejności od materiału znajdującego się najbardziej na spodzie do materiału umieszczonego najbardziej na wierzchu)

$$X^* \mathrel{-}= \sum_{i=0}^{nL} n_{S.embedded.L,i} \cdot X_{L,i} \quad (92)$$

$$X^* \mathrel{-}= \sum_{i=0}^{nFG} n_{S.embedded.FG,i} \cdot X_{FG,i} \quad (93)$$

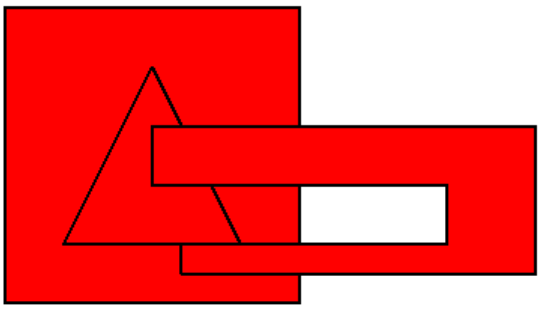
Stosując powyższy algorytm i wstawiając za X odpowiednio A, S_x, S_y można uzyskać ważone pole powierzchni przekroju oraz ważone momenty statyczne. W dalszej kolejności otrzymuje się niemal wprost ze wzoru (94) ważony środek geometryczny przekroju.

$$C^* = \left(x_c = \frac{S_y^*}{A^*}, y_c = \frac{S_x^*}{A^*}\right) \quad (94)$$

Znalezienie środka geometrycznego C^* jest istotne gdyż stanowi on początek układu współrzędnych XC^*Y w którym będą wyznaczane siły przekrojowe N, M_x, M_y na dalszych etapach obliczeń. Układ ten jest układem centralnym przekroju zespolonego. W algorytmie generowania profili odkształceń będzie on obracany o 360 stopni względem punktu C^* w celu wyznaczenia wszystkich możliwych położeń profili odkształceń w odniesieniu do przekroju zespolonego.

4.5 Generowanie kombinacji przecięć komponentów powierzchniowych

Definiowany przekrój zespolony może posiadać dowolną liczbę komponentów powierzchniowych. Komponenty te nachodząc na siebie tworząc przecięcia. Podczas generowania wszystkich możliwych przecięć można przyjąć, że są one wielokątami tj. należy uprzednio dokonać linearyzacji łuków. Ze względu na dowolny kształt poszczególnych powierzchni w wyniku przecięcia dwóch wielokątów może powstać więcej niż jeden nowy wielokąt. Możliwe są również sytuacje w których nachodzi na siebie wiele warstw stąd w celu jak najbardziej ogólnego potraktowania problemu należy znaleźć przecięcia nie tylko pomiędzy dwoma komponentami, ale dowolną liczbą *n*-komponentów.



Rys. 29 Abstrakcyjny przypadek przecinania się trzech komponentów powierzchniowych

Zakładając, że poszczególne komponenty powierzchniowe na rysunku (Rys. 29) numerowane są począwszy od materiału znajdującego się jak najbardziej na spodzie: kwadrat S_1 , trójkąt S_2 , ceownik S_3 należy rozpatrzeć następujące przypadki przecięć. $-n_1 S_2 \cap S_1$, $-n_2 S_3 \cap S_2$, $-n_1 S_3 \cap S_1$, $+n_1 S_3 \cap S_1 \cap S_2$. Znaki przed wyrażeniami oznaczają, że przecięcia wpływa na charakterystykę przekroju zespolonego dodatnio lub ujemnie. Rozpatrując przecięcia dla większej liczby komponentów można dostrzec zależność, że każdy *k*-ty komponent przecina się z od *1* do *(k-1)* innymi komponentami. Powierzchnie z którymi przecina się *k*-ty komponent mogą być w łatwy sposób wygenerowane stosując kombinacje bez powtórzeń i elementów ze zbioru *(k-1)* elementowego C_i^{k-1} . Należy zauważyć, że i może być dowolną liczbą z przedziału $i \in [1, k-1]$. Oznacza to, że dla k-tej powierzchni należy wygenerować odpowiednio następujące kombinacje: $C_1^{(k-1)}$ (*(k-1)* kombinacji), $C_2^{(k-1)}$ ($\frac{(k-1)!}{2(k-3)!}$ kombinacji), …. , $C_{(k-1)}^{(k-1)}$ (1 kombinacja). Liczba poszczególnych kombinacji wynika ze wzoru na symbol newtona (95).

$$C_s^r = \left(\frac{r}{s}\right) = \frac{r!}{s! \cdot (r-s)!} \quad (95)$$

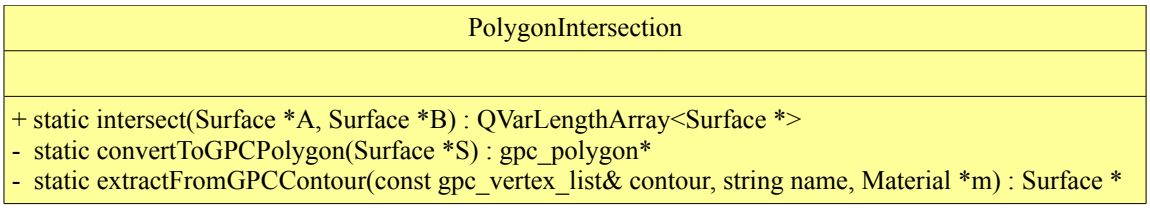
C_s^r oznacza kombinacje bez powtórzeń *s*-elementowe ze zbioru *r*-elementowego. Rozpatrując wszystkie możliwe przecięcia dla n-wielokątów należy iterować stosowanym wcześniej parametrem *k* (określającym numer aktualnie rozpatrywanego komponentu) w przedziale $k \in [1, n]$. Dla przykładu z rys. 29 można by zapisać to w następujący sposób:

$$S_1 ---> \; k=1 ----> 0$$
$$S_2 ---> k=2 ----> C_1^1 ---> \{S_1\}$$
$$S_3 ---> \; k=3 ----> C_1^2, C_2^2 ---> \{S_1\}, \{S_2\}, \{S_1, S_2\}$$

Warto zwrócić również uwagę na wagi n_i wynikające z uwzględnianego w danej kombinacji

materiału oraz stojące przed nimi znaki „+/-”. Waga $n_i = E_i / E_{ref}$ określana jest na podstawie materiału przypisanego do pierwszej powierzchni z którą przecina się *k*-ty komponent, przy założeniu, że wygenerowana kombinacja powierzchni została uporządkowana względem rosnących numerów. Znak „+/-” kombinacji zależy natomiast od liczby wygenerowanych powierzchni z którymi ma przeciąć się *k*-ty komponent, jeżeli liczba ta jest parzysta to przyjmuje się znak „+”, jeżeli jest nieparzysta to znak „-”.

W kodzie do generowania kombinacji używa się obiektów klasy `CombinationGenerator`, której konstruktor przyjmuje jako parametr liczbę zdefiniowanych powierzchni w przekroju. Wszystkie wygenerowane kombinacje można natomiast pobrać za pomocą metody `vector< vector<int> > getAllCombinations(void)`. Generowanie kombinacji przecięć powierzchni w celu poprawnego uwzględnienia wpływów od tych przecięć stosowane jest nie tylko w przypadku charakterystyk geometrycznych A^*, S_x^*, S_y^*, ale również w przypadku wyznaczania sił przekrojowych N, M_x, M_y. Do wyznaczania wpływów od tych przecięć stosowane są prywatne metody klasy `Surface` należące do rodziny `double calculateIntersectionXXX(...)`. Rzeczywiste przecięcia wielokątów (komponentów powierzchniowych) są innymi wielokątami (komponentami powierzchniowymi). Stąd na przecięciu można dokonywać tych samych obliczeń jak na zwykłym komponencie powierzchniowym tj. wyznaczać jego charakterystyki geometrzyczne oraz całkowań naprężenia po powierzchni przecięcia w celu poszukiwania sił przekrojowych.
Do wyznaczania przecięć w realizowanym programie została wykorzystana biblioteka GPC (ang. General Polygon Clipping Library) udostępniana darmowo dla celów niekomercyjnych przez University of Manchester Advanced Interfaces Group. Umożliwia ona wyznaczanie przecięć pomiędzy dwoma dowolnymi wielokątami, które mogą być wklęsłe lub wypukłe, wewnętrznie przecinające się, zawierać otwory lub składać się z rozdzielnych konturów. Biblioteka wykorzystuje do znajdowania przecięć rozszerzony algorytm Vatti'ego (ang. Vatti clipping algorithm). W kodzie realizowanego programu biblioteka GPC została opakowana klasą PolygonIntersection (Rys. 30).



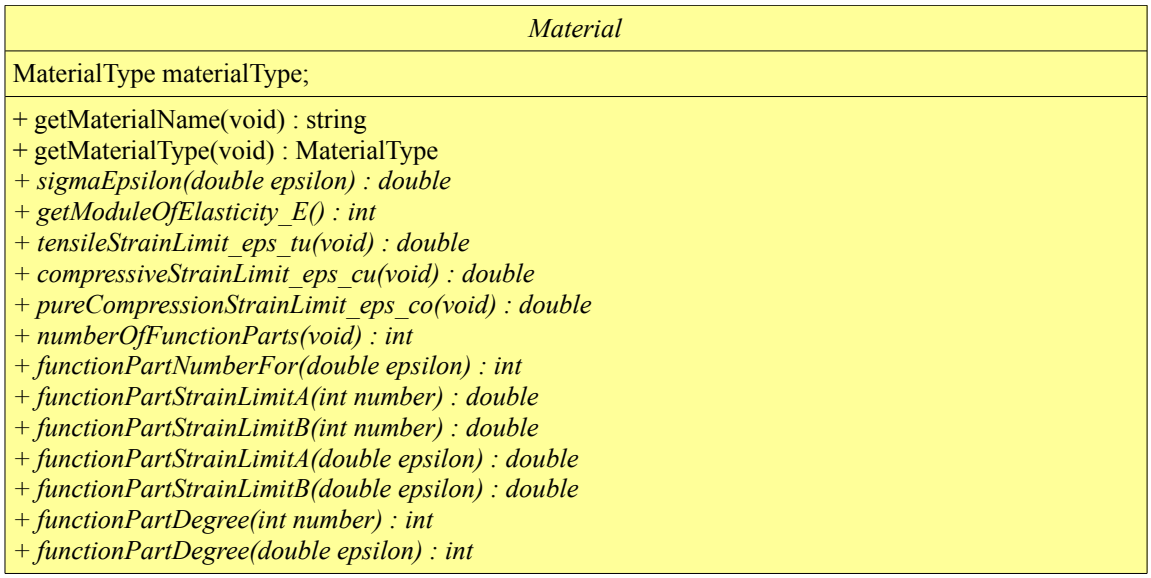
Rys. 30 Diagram UML klasy PolygonIntersection opakowującej bibliotekę GPC.

5. NUMERYCZNY ZAPIS ZWIĄZKÓW KONSTYTUTYWNYCH MATERIAŁÓW

W rozdziale 2.4 niniejszej pracy przedstawiono wymagania normowe (zgodnie z PN-EN 1994-1-1) [3] dotyczące materiałów stosowanych na konstrukcje zespolone tj. beton, stal zbrojeniowa i stal konstrukcyjna. W realizowanym programie predefiniowano parametry materiałowe i związki konstytutywne bazując na tych wymaganiach. Algorytm służący do obliczania nośności przekrojów zespolonych powinien uwzględniać możliwość zdefiniowania jak najbardziej dowolnych związków konstytutywnych dla materiałów. Zależności $\sigma - \epsilon$ mają najczęściej postać funkcji odcinkowo wielomianowych. Charakter tych funkcji należy uwzględnić już na etapie definiowania materiałów, tak aby umożliwić w przyszłości łatwą rozbudowę ich bazy lub wyposażyć program w możliwość tworzenia własnych definicji materiałów przez użytkowników. Punkty charakterystyczne funkcji $\sigma - \epsilon$ wyznaczające przedziały jej ciągłości $f_i(\epsilon)$ to wartości graniczne odkształceń: ϵ_{tu} (odkształcenie przy granicznej wytrzymałości na rozciąganie), ϵ_{co} (odkształcenia przy wytrzymałości na czyste ściskanie), ϵ_{cu} (odkształcenia przy granicznej wytrzymałości na ściskanie).

5.1 Klasa abstrakcyjna Material

Bazę całego modułu definiowania materiałów stanowi klasa abstrakcyjna `Material` (Rys. 31).



Rys.31 Diagram UML klasy abstrakcyjnej Material

Dodatkowo w kodzie zdefiniowano dwa wyjątki rozszerzające `std::exception`: `MaterialNotFoundException` oraz `StrainOutOfBoundsException`. Pierwszy z nich chroni przed próbą użycia niezdefiniowanego materiału, drugi wyrzucany jest w przypadku żądania wartości naprężeń dla odkształceń znajdujących się poza zakresem dopuszczalnym dla danego materiału. Klasa rozszerzająca typ abstrakcyjny `Material` musi zaimplementować następujące metody wirtualne idąc od góry w przedstawionym diagramie ULM (Rys. 31):

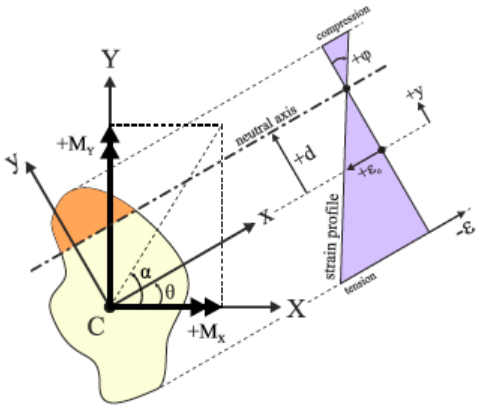
- funkcje $\sigma-\epsilon$ obliczającą wartość naprężeń w zależności od wartości odkształceń
- metodę zwracającą wartość modułu sprężystości E, który może być zdefiniowany jako stała np. dla stali E_s lub wyznaczany z zależności np. dla betonu $E_{cm}=E(f_{cm}(f_{ck}))$.
- trzy funkcje zwracające wspomniane już wartości graniczne odkształceń: $\epsilon_{tu}, \epsilon_{cu}, \epsilon_{co}$.
- metody przydatne przy całkowaniu numerycznym Gauss-Legendre z wykorzystaniem mapowania odkształceń (ang. adaptive strain-mapped Gaussian integration). Są to funkcje które zwracają liczbę przedziałów ciągłości (części) zależności $\sigma-\epsilon$, numer i konkretnej funkcji f_i dla zadanej wartości odkształceń, granice dolne (bardziej rozciągane) i górne (bardziej ściskane) odkształceń dla poszczególnych f_i oraz w zależności od ϵ. Dwie ostatnie metody zwracające stopień wielomianu dla danego fragmentu f_i lub zadanych odkształceń pozwalają na dostosowanie liczby punktów Gaussa używanych w całkowaniu numerycznym celem otrzymania dokładnych wyników przy jednoczesnej optymalizacji obliczeń.

Predefiniowane klasy rozszerzające typ `Material` to `Concrete`, `StracturalSteel`, `Reinforcement`, `Hollow` oraz dwie wykorzystane dla celów testowych `FafitisConcrete`, `FafitisReinforcement` zawierające definicję materiałów użytych w kilku benchmarkach, bazujące na amerykańskiej normie ACI 318-08 [16] oraz anglosaskim systemie metrycznym. Każda z powyższych klas definiuje stałe materiałowe, implementuje metody wirtualne oraz metody pomocnicze np. zwracające wskaźniki na metody swojego obiektu implementujące dany fragment zależności $\sigma-\epsilon$ tj. f_i czy wyznaczające inne charakterystyki materiałowe na podstawie zdefiniowanych stałych. Na szczególną uwagę zasługuje klasa `Hollow`, która definiuje fikcyjny materiał stosowany dla otworów, zwraca ona $E=0$, a dla pozostałych metod wirtualnych

wyrzuca wyjątek `HollowCalculationAttemptException`.

6. CAŁKOWANIE NUMERYCZNE NAPRĘŻĘŃ PO POWIERZCHNI PRZEKROJU

W celu otrzymania sił przekrojowych N, M_x, M_y dla zadanego profilu odkształceń oraz kąta jego obrotu względem przekroju (Rys. 32) należy scałkować odpowiadające temu profilowi naprężenia po powierzchni poszczególnych komponentów przekroju zespolonego.



Rys. 32 Przykładowy profil odkształceń obrócony o kąt θ względem przekroju oraz układy osi centralnych globalnych i lokalnych [1]

Metoda wyznaczania sił wewnętrznych jest odmienna w zależności od typu komponentu: powierzchniowy (ang. surface), liniowy (ang. line), zbrojenie (ang. fibergroup). Globalne wartości N, M_x, M_y wyznaczamy jako sumę wpływów od poszczególnych komponentów, uwzględniając przy tym przecięcia komponentów powierzchniowych i osadzenie elementów liniowych lub zbrojenia w innym elemencie. Wpływ od przecięć jest odpowiednio odejmowany lub dodawany zgodnie z kombinacjami generowanymi w sposób analogiczny jak przy wyznaczaniu charakterystyk geometrycznych (rozdz. 4.5). Natomiast w odniesieniu do elementów liniowych i zbrojenia należy odjęć wpływ od materiałów w którym są one osadzone. Przyjmuje się tutaj dla uproszczenia obliczeń założenie, że komponenty te mogą być osadzone tylko w jednym materiale. W kodzie powyższe zadanie realizują metody `Section::summateInternalActions_XXX` oraz `Section::calculateIntersectionXXX`, gdzie za *XXX* możemy podstawić odpowiednio symbole N, M_x, M_y . Dodatkowo należy zwrócić uwagę na fakt, że wartości momentów M_x, M_y zwracane przez te metody wyrażone są w lokalnym układzie współrzędnych obróconym względem układu globalnego o kąt θ zgodny z kątem obrotu profilu odkształceń w odniesieniu do przekroju. W celu otrzymania odpowiadających im wartości wyrażonych w układzie globalnym można skorzystać ze wzorów transformacyjnych (96), (97). W kodzie zostały one zaimplementowane za pomocą metod statycznych `Section::transferBackFromLocalMy`, `Section::transferBackFromLocalMx`.

$$M_X = M_{x,loc} \cdot \cos(-\theta) + M_{y,loc} \cdot \sin(-\theta) \quad (96)$$

$$M_y = -M_{x,loc} \cdot \sin(-\theta) + M_{y,loc} \cdot \cos(-\theta) \quad (97)$$

6.1 Całkowanie numeryczne dla komponentów powierzchniowych

Siły przekrojowe $N_{A,i}, M_{x,A,i}, M_{y,A,i}$ dla zadanego komponentu można wyznaczyć z zależności odpowiednio (49), (50), (51) przedstawionych w rozdziale 2.5. Omówiona tam zamiana całki powierzchniowej na krzywoliniową po obwodzie komponentu powierzchniowego pozwala na wykonanie obliczeń stosując jedną z powszechnie znanych metod całkowania numerycznego

przedstawionych w rozdziale 2.6. W realizowanym programie celem uzyskania jak największej dokładności obliczeń wykorzystano metodę Gaussa-Legendre z mapowaniem granicznych odkształceń na współrzędne y. Zależnością wyjściową do otrzymania dowolnej siły wewnętrznej jest formuła (52) uzyskana w wyniku zastosowania twierdzenia Greena do całki powierzchniowej. Symbol $R_{A,i}$ we wzorze oznacza jedną z trzech poszukiwanych wartość sił przekrojowych zależnie od parametrów (r,s), są to odpowiednio $N_{A,i}$ dla *(r,s) = (0,0)*, $M_{x,A,i}$ dla *(r,s) = (0,1)* oraz $-M_{y,A,i}$ dla *(r,s) = (1,0)*. Całkę krzywoliniową po obwodzie C_i komponentu powierzchniowego we wzorze (52) można zapisać jako sumę całek liniowych po kolejnych krawędziach wieloboku (98), biorąc pod uwagę, że zdefiniowane w programie powierzchnie są linearyzowane właśnie do wielokątów.

$$R_{A,i}=\frac{1}{r+1}\oint_{Ci} x^{r+1}\cdot y^{s}\cdot\sigma(y)\,dy=\frac{1}{r+1}\sum_{j=0}^{nP.i-1}\oint_{segment,\,j} x^{r+1}\cdot y^{s}\cdot\sigma(y)\,dy=\frac{1}{r+1}\sum_{j=0}^{nP.i-1} S_j \quad (98)$$

W powyższym wzorze *nP,i* wyraża liczbę wierzchołków i-tej powierzchni A_i , a tym samym liczbę krawędzi tego wieloboku. Natomiast S_j jest wspominaną już całką liniową po j-tej krawędzi. W kodzie klasa Surface posiada trzy metody pozwalające wyznaczyć siły przekrojowe reprezentowanego przez jej obiekt komponentu powierzchniowego: `internalAction_My()`, `internalAction_My()`, `internalAction_My()`. Metody te jako jedyny parametr pobierają obiekt klasy `StrainProfile` (definiującej profil odkształceń) i opakowują wywołanie uniwersalnej procedury całkowania naprężeń po powierzchni komponentu `performStressIntegration()`. Funkcja ta jest odpowiedzialna za dobór współczynników *(r,s)* w zależności od poszukiwanej siły przekrojowej oraz wykonanie sumowania całek liniowych S_j zgodnie z (98).

```
double Surface::performStressIntegration(StrainProfile *profile,
                                         InternalAction actionType)
{
      double internalAction = 0.0;
      int r = 0, s = 0;
      if(actionType == ACTION_N){
         r = 0; s = 0;
      } else if(actionType == ACTION_Mx) {
         r = 0; s = 1;
      } else if(actionType == ACTION_My) {
         r = 1; s = 0;
      }
      double xi,yi, xi_1, yi_1;
      int numOfPoints = this->numberOfPoints();
      Point *sectionCentroid = profile->getSection()->getGeometricalCenter();
      double omega = profile->getRotationAngle();
      for(int i=0; i< numOfPoints; i++) {
        xi = points[i]->getX(); yi = points[i]->getY();
         if( i+1 < numOfPoints) {
              xi_1 = points[i+1]->getX(); yi_1 = points[i+1]->getY();
         } else {
              xi_1 = points[0]->getX(); yi_1 = points[0]->getY();
         }
         Point pi(xi,yi), pi_1(xi_1, yi_1);
         pi.setOriginAndRotation(sectionCentroid->getX(),
                                 sectionCentroid->getY(), omega);
         pi_1.setOriginAndRotation(sectionCentroid->getX(),
                                   sectionCentroid->getY(), omega);
         Segment lj(pi,pi_1, std::to_string(i));
         internalAction += segmentIntegral_Ij(lj, profile, r, s);
      }
      internalAction *= (double) 1/(r+1);
      return internalAction;
}
```

Rys. 33 Procedura `Surface::performStressIntegration`

Całkę liniową S_j z (98) można uprościć do całki liniowej z funkcji jednej zmiennej $F_j(y)$ (54) stosując podstawienie (100) za *x*. Taką całkę można już obliczać stosując metodę Gaussa-

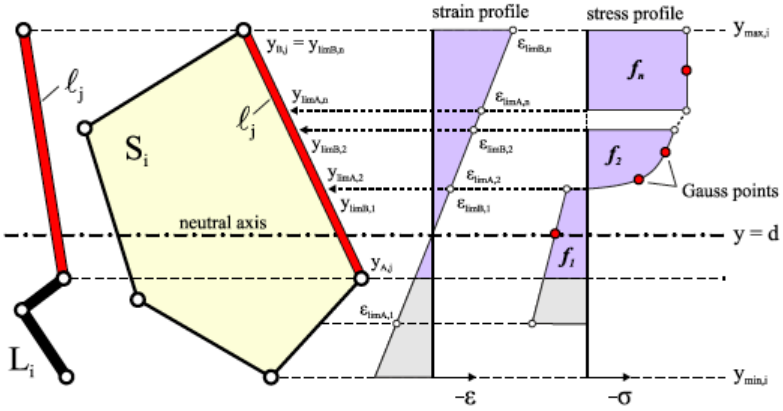
Legendre, która daje dokładne wyniki dla *nG* punktów Gaussa zależnych od stopnia wielomianu całkowanej funkcji (101).

$$S_j = \oint_{segment,\,j} F_j(y)\,dy = \oint_{segment,\,j} (a_j \cdot y + b_j)^{r+1} \cdot y^s \cdot \sigma(y)\,dy \quad (99)$$

$$x = a_j \cdot y + b_j \quad (100)$$

$$nG = n \setminus 2 + 1 \quad (101)$$

Warto zauważyć, że dla danej krawędzi *j* wieloboku wykres naprężeń $\sigma(\epsilon)$ odpowiadający profilowi odkształceń może nie być funkcją ciągła tj. może składać się z kilku fragmentów f_k. W celu uwzględnienia tego faktu w obliczeniach należy zastosować mapowanie odkształceń granicznych $\epsilon_{\lim,k}$ dla poszczególnych f_k na współrzędne $y_{\lim,k}$ rozpatrywanej krawędzi (102). Jeżeli odwzorowane wartości współrzędnych $y_{\lim,k}$ wypadają poza współrzędnymi $y_{A,j}, y_{B,j}$ końców j-tej krawędzi to należy je odpowiednio dostosować tak by należały do przedziału domkniętego $[y_{A,j}, y_{B,j}]$. Następuje więc podział krawędzi na *nf* pododcinków, którym odpowiadają jednorodne zależności naprężenie-odkształcenie $f_k(\epsilon)$ (Rys. 34).



Rys. 34 Odwzorowanie odkształceń granicznych f_k na współrzędne $y_{\lim,k}$ [1]

$$y = \frac{(\epsilon_0 - \epsilon)}{\phi} \quad (102)$$

Jak już wspomniano w rozdziale 5.1 każdy materiał posiada metody, które umożliwiają pobranie liczby funkcji f_k oraz ich odkształcenia graniczne $\epsilon_{\lim,A,k}, \epsilon_{\lim,B,k}$ i stopień *n* ich wielomianu. Wobec powyższej uwagi całkę liniową funkcji jednej zmiennej we wzorze (99) można zapisać jako sumę całek oznaczonych na przedziałach $[y_{\lim,A,k}, y_{\lim,B,k}]$ (103).

$$S_j = \oint_{segment,\,j} F_j(y)\,dy = \sum_{k=0}^{nf-1} \int_{y\_\lim,A,k}^{y\_\lim,B,k} F_j(y)\,dy \quad (103)$$

Do całki oznaczonej znajdującej się pod sumą we wzorze (103) można w sposób bezpośredni zastosować całkowanie Gaussa (62) otrzymując (104), gdzie y_m jest *m*-tym punktem Gaussa (dla *nG* przyjętych) po transformacji (61) z przedziału [-1,1] do przedziału $[y_{\lim,A,k}, y_{\lim,B,k}]$ (105).

$$S_j = \sum_{k=0}^{nf-1} \left(\frac{1}{2} \cdot (y_{\lim,B,k} - y_{\lim,A,k}) \cdot \sum_{m=0}^{nG-1} w_m \cdot F_j(y_m)\right) \quad (104)$$

$$y_m = \frac{(y_{\lim,B,k} + y_{\lim,A,k})}{2} + \frac{\xi_m \cdot (y_{\lim,B,k} + y_{\lim,A,k})}{2} \quad (105)$$

Wartości (w_m, ξ_m) dla *nG* przyjętych punktów Gaussa, można odczytać z (Tablica 2). W kodzie programu do generowania punktów Gaussa i odpowiadających im wag wykorzystano obiekt klasy GaussLegendre, która przechowuje je w wewnętrznej tablicy. Fragment kodu z procedury całkowania j-tej krawędzi powierzchni przedstawiono na (Rys. 35).

```
double Surface::segmentIntegral_Ij(const Segment &lj,
            StrainProfile *profile, double r, double s){
 //xi = …, yi = …, xi_1 = …, yi_1 = …, aj = …, bj = …,
 //eps_o = … , fi = …, y_Aj = …, y_Bj=..., internalAction - 0.0
 for(int k=0; k < material->numberOfFunctionParts(); k++) {
       //y_limA_k = ..., y_limB_k = ...
       double Ij_k = 0.5*(y_limB_k - y_limA_k);
       int fkOrder = material->functionPartDegree(k);
       int nG = (fkOrder + r + 1 + s)/2 + 1;
       double summation = 0.0;
       GaussLegendre gauss(nG);
       for(int m = 0; m < nG; ++m) {
             double ym = 0.5*(y_limB_k + y_limA_k) +
                                 gauss.lambda(m)*0.5*(y_limB_k - y_limA_k);
             double eps_ym = eps_o - fi*ym;
             double Fj_ym  = pow(aj*ym + bj, r+1)*pow(ym,s)
                                          *material->sigmaEpsilon(eps_ym);
             //adding up consecutive parts of Gaussian quadrature
       summation += gauss.weight(m)*Fj_ym; //N -> [N/mm] Mx, My -> [N]
       }
       Ij_k *= summation;
       internalAction += Ij_k;
 }
}
```

Rys. 35 Procedura `Surface::segmentIntegral_Ij`

6.2 Całkowanie numeryczne dla komponentów liniowych

Siły przekrojowe $N_{L,i}, M_{x,L,i}, M_{y,L,i}$ dla *i*-tego komponentu liniowego zdefiniowane są w analogiczny sposób za pomocą całki krzywoliniowej (106). Element liniowy jest łamaną składającą się z *nP,i* odcinków, stąd całkę krzywoliniową można wyrazić jako sumę całek S_j po każdym z tych odcinków (107). Całka liniowa S_j może być wyrażona jako całka liniowa jednej zmiennej stosując podobnie jak w przypadku komponentów powierzchniowych podstawienie za *x* (100). Warto zwrócić uwagę na parametr t_j oznaczający grubość *j*-tego odcinka wyznaczany ze wzoru (108). W odniesieniu do całki S_j podobnie jak w przypadku elementów powierzchniowych stosuje się mapowanie odkształceń (102) i podział odcinków na *nf* pododcinków odpowiadających poszczególnym fragmentom f_k zależności naprężenie-odkształcenie. Dla każdego pododcinka otrzymuje się w ten sposób całkę oznaczoną wyrażoną w granicach $[y_{\lim,A,k}, y_{\lim,B,k}]$ (109). Całkę tę można obliczyć stosując metodę Gaussa-Legendre (111) i przyjmując w zależności od stopnia wielomianu f_k *nG* punktów Gaussa (101).

$$R_{L,i} = \oint_{C,i} x^r y^s t(y) \sigma(y) dy \quad (106)$$

$$R_{L,i} = \sum_{j=0}^{nP,i-1} S_j = \sum_{j=0}^{nP,i-1} \oint_{segment,j} x^r y^s t_j \sigma(y) dy \quad (107)$$

$$t_j = \frac{A_j}{(\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2})} \quad (108)$$

$$S_j = \oint_{segment,j} F_j(y)\,dy = \sum_{k=0}^{nf-1} \int_{y\_\lim,A,k}^{y\_\lim,B,k} F_j(y)\,dy \quad (109)$$

$$F_j(y) = (a_j y + bj)^r \cdot y^s t_j \sigma(y) \quad (110)$$

$$S_j = \sum_{k=0}^{nf-1} \left(\frac{1}{2} \cdot (y_{\lim,B,k} - y_{\lim,A,k}) \cdot \sum_{m=0}^{nG-1} w_m \cdot F_j(y_m)\right) \quad (111)$$

Warto zauważyć, że w przypadku odcinków poziomych, gdzie $y_{\lim,A,k} = y_{\lim,B,k}$ ich wpływ na siłę przekrojową zostanie pominięty. Stąd należy uwzględnić odcinki poziome w obliczeniach za pomocą wzorów (112), (113), (114).

$$N_j = \sigma(y_j) \cdot A_j \quad (112)$$

$$M_{x,j} = \sigma(y_j) \cdot y_j \cdot A_j \quad (113)$$

$$M_{y,j} = \sigma(y_j) \cdot \frac{(x_j + x_{j+1})}{2} \cdot A_j \quad (114)$$

W kodzie siły przekrojowe dla danego komponentu liniowego można otrzymać za pomocą metod `Line::internalAction_XXX()`, gdzie za *XXX* należy podstawić odpowiednio N, M_x, M_y. Podobnie jak w klasie Surface opakowują one wywołanie metody uniwersalnej `Line::performStressIntegration()`. Procedura ta jest analogiczna jak przedstawiona na (Rys.33) tj. w pętli sumuje wyniki całkowań dla poszczególnych odcinków elementu liniowego wywołując funkcje `Line::segmentIntegral_Ij()`. Istotną różnica jest fakt, że końcowa wartość nie jest przemnażana prze $1/(r+1)$.

6.3 Wyznaczenie sił przekrojowych dla zbrojenia

W przypadku zbrojenia (ang. fibergroup) wyznaczenie sił wewnętrznych jest trywialne. Siły przekrojowe *i*-tego komponentu można wyznaczyć wprost ze wzoru (115).

$$R_{FG,i} = \sum_{j=0}^{nP,i-1} \left(x_j^r y_j^s \sigma(\epsilon_j(y_j))\, A_j\right) \quad (115)$$

Jest to po prostu suma wpływów od *nP,i* prętów zbrojeniowych zdefiniowanych w komponencie. Wyrażenie pod sumą oznacza iloczyn naprężenia (odpowiadającego odkształceniu na wysokości środka ciężkości pręta zbrojeniowego) i pola powierzchni przekroju pręta dla siły osiowej *N*, dla momentu M_x iloczyn ten jest dodatkowo przemnożony przez y_j, a dla M_y przez x_j.
Wartość odkształceń $\epsilon_j(y_j)$ można obliczyć stosując formułę (116), gdzie ϵ_o to odkształcenia na wysokości początku lokalnego układu współrzędnych, a ϕ kąt nachylenia wykresu odkształceń (krzywizna).

$$\epsilon_j(y_j) = \epsilon_o - \phi\, y_j \quad (116)$$

W kodzie powyższe zależności zostały zdefiniowane w metodzie `FiberGroup::performStressIntegration()`, która podobnie jak w przypadku innych komponentów opakowana jest przez funkcje `FiberGroup::internalAction_XXX()`, gdzie za *XXX* można podstawić odpowiednio N, M_x, M_y.

7. GENEROWANIE PROFILI ODKSZTAŁCEŃ GRANICZNYCH

Metody `summateInternalActions_XXX()` oraz `internalAction_XXX()` służące do wyznaczania sił przekrojowych odpowiednio dla całego przekroju i dla poszczególnych jego komponentów jako jedyny parametr pobierają wskaźnik na obiekt klasy StrainProfile. Klasa ta służy do definiowania profili odkształceń, które mają być poddane procedurze całkowania numerycznego. Jej konstruktor przyjmuje następujące parametry: przekrój (wskaźnik na typ Section *) , kąt θ obrotu profilu odkształceń względem przekroju przekazanego jako pierwszy argument, wartość odkształceń ϵ_{lower} dla y_{MIN} oraz ϵ_{upper} dla y_{MAX} zdefiniowanych w lokalnym centralnym układzie współrzędnych obróconym o kąt przekazany jako drugi argument. Na podstawie tych danych wyliczane są pozostałe parametry profilu odkształceń tj. ϕ, odległość pionowa osi obojętnej od początku lokalnego układu współrzędnych d oraz odkształcenia na wysokości początku układu współrzędnych ϵ_o . Mając zdefiniowane te parametry można z łatwością wyznaczyć wartości odkształceń (116) i co za tym idzie naprężeń w dowolnym punkcie przekroju zespolonego. Równie dobrze możemy obliczyć dla jakiej wartości współrzędnej y profil osiąga zadane odkształcenia jak to ma miejsce w przypadku mapowania odkształceń (102). Dla pojedynczego profilu odkształceń można wyznaczyć dokładnie jeden punkt w przestrzeni (M_x, M_y, N). W celu określenia nośności przekroju zespolonego oraz skonstruowania krzywych i powierzchni interakcji potrzebna jest stosunkowo gęsta siatka punktów odpowiadających granicznym profilom odkształceń tj. takim profilom w przypadku których przynajmniej w jednym punkcie wykres odkształceń osiąga wartość graniczną np. graniczne ściskanie lub graniczne rozciąganie. Profile to można generować metodami iteracyjnymi w zakresie dopuszczalnych granicznych odkształceń ściskających i rozciągających dla całego przekroju. Przez odkształcenia dopuszczalne graniczne w przekroju zespolonym rozumieć będziemy trzy punkty $(\epsilon_{tu}, y_{tu}), (\epsilon_{cu}, y_{cu}), (\epsilon_{co}, y_{co})$ dla których podczas generowania profili odkształceń nie dojdzie do przekroczenia odkształceń granicznych żadnego z komponentów składowych.

7.1 Wyznaczenie dopuszczalnych odkształceń granicznych

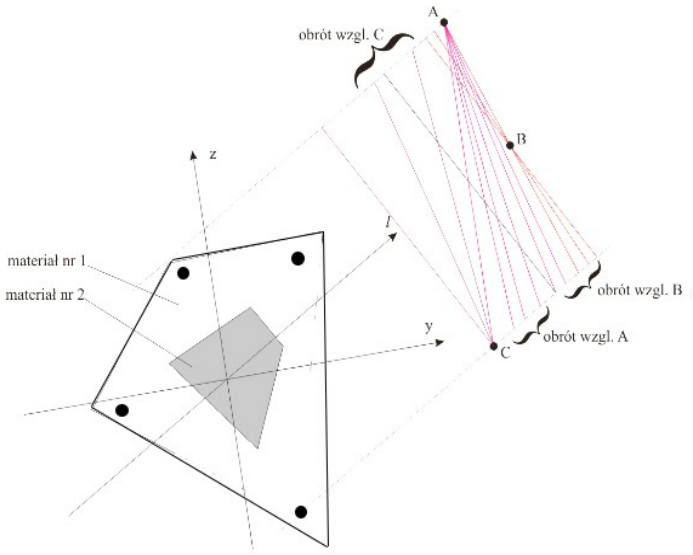
Jak już wspomniano punkty te mają definiować pewien obszar w którym wygenerowane w sposób iteracyjny profile odkształceń nie będą przekraczać odkształceń granicznych w żadnym z materiałów (komponentów). Jako sposób wyznaczenia tych punktów zaproponowano następujący algorytm:

1. Przyjąć ustaloną wartość kąta θ obrotu profili odkształceń względem przekroju zespolonego
2. Przejść pętlą po wszystkich komponentach przekroju (powierzchnie, linie, zbrojenie), które dopuszczają odkształcenia rozciągające > 0 (nie równe) np. stal konstrukcyjna, zbrojenie, etc. Jeżeli taki materiał nie występuje w przekroju, np. składa się on tylko z elementów betonowych to zakładamy $(\epsilon_{tu}=0, y_{tu}=y_{MIN})$
3. Dla *i*-tego komponentu rozciąganego z pkt.2 przejść pętlą zagnieżdżoną po wszystkich elementach (powierzchniach, liniach, zbrojeniu) przekroju dopuszczających odkształcenia ściskające < 0. Należy przy tym pominąć elementy dla których $y_{max\_j} \le y_{min\_i}$ w lokalnym centralnym układzie współrzędnych. Oznacza to przyjęcie założenia, że ściskanie występuje na górze przekroju natomiast rozciąganie na dole przekroju.
4. Dla każdego *i*-tego komponentu rozciąganego z pkt. 2 i *j*-tego komponentu ściskanego z pkt. 3 na podstawie odpowiadających im punktów $(\epsilon_{tu\_i}, y_{tu\_i}), (\epsilon_{cu\_j}, y_{cu\_j})$ należy wyznaczyć prostą przechodzącą przez te punkty oraz jej przecięcia z prostymi $y=y_{MAX}, y=y_{MIN}$.
5. Otrzymane w pkt. 4 punkty $(\epsilon_{tu\_MIN\_ij}, y_{MIN}), (\epsilon_{cu\_MAX\_ij}, y_{MAX})$ należy porównać z bieżącymi punktami $(\epsilon_{tu\_MIN}, y_{MIN}), (\epsilon_{cu\_MAX}, y_{MAX})$ (początkowo przyjęto odkształcenia równe 0 w tych punktach). Jeżeli nowe wartości dla *i*-tego materiału rozciąganego i *j*-tego ściskanego limitują odkształcenia graniczne w przekroju lub wartości dotychczasowych odkształceń $\epsilon_{tu\_MIN}, \epsilon_{cu\_MAX}$ są równe 0 to należy nadpisać bieżące wartości odkształceń granicznych nowy wyliczonymi dla *ij*.

Należy zapamiętać również punkty $(\epsilon_{tu\_i}, y_{tu\_i}), (\epsilon_{cu\_j}, y_{cu\_j})$ w zmiennych $\epsilon_{tu}, y_{tu}, \epsilon_{cu}, y_{cu}$.
Za limitujące przyjmuje się te wartości odkształceń na dolnej i górnej krawędzi przekroju zespolonego, których wartości bezwzględne są mniejsze tj. leżą bliżej pionowej osi profilu odkształceń. Należy uwzględniać jedynie ujemne odkształcenia ściskające i dodatnie odkształcenia rozciągające. Warunek ten powinien zostać spełniony poprzez wcześniejsze założenie $y_{max\_j} > y_{min\_i}$.
6. Dla każdych nowo wybrany odkształceń granicznych $\epsilon_{tu\_MIN}, \epsilon_{cu\_MAX}$ na dolnej i górnej krawędzi przekroju zespolonego należy dodatkowo sprawdzić czy nie powodują one przekroczenia dla y_{co_j} odkształceń granicznych przy czystym ściskaniu *j*-tego komponentu $(\epsilon_{co\_j}, y_{co\_j})$.
Jeżeli doszło do przekroczenia odkształceń ϵ_{co_j} wtedy stają się one limitującymi odkształceniami ściskającymi w przekroju i należy wyznaczyć nowe odkształcenia ϵ_{cu}, y_{cu} jako punkty przecięcia prostej przechodzącej przez punkty $(\epsilon_{tu\_j}, y_{tu\_j}), (\epsilon_{co\_j}, y_{co\_j})$ oraz $y = y_{cu\_j}$.
7. Ostatecznie otrzymujemy trzy pary wartości ograniczające graniczne odkształcenia w całym przekroju zespolonym $(\epsilon_{tu}, y_{tu}), (\epsilon_{cu}, y_{cu}), (\epsilon_{co}, y_{co})$. Są to odpowiednio odkształcenia rozciągające, odkształcenia ściskające i odkształcenia przy czystym ściskaniu oraz odpowiadające im wartości współrzędnych *y* w których te ograniczenia odkształceń są nakładane.

W programie powyższy algorytm został zaimplementowany w klasie `SectionSolver` dziedziczącej po klasie nadrzędnej `Section` w metodzie `calculateTensileAndCompressiveStrainLimits()`.

7.2 Sterowanie profilem odkształceń w zakresie dopuszczalnych odkształceń granicznych



Rys. 36 Sterowanie profilem odkształceń w zakresie odkształceń granicznych.

Dopuszczalne odkształcenia graniczne $(\epsilon_{tu}, y_{tu}), (\epsilon_{cu}, y_{cu}), (\epsilon_{co}, y_{co})$ wyznaczone za pomocą algorytmu z (Rozdział 7.1) to odpowiednio punkty C, A, B na (Rys. 36). Sterowanie profilem odkształceń polega na generowaniu w sposób iteracyjny kolejnych profili granicznych odkształceń w ramach obszaru wyznaczanego przez punkty C, A, B. Algorytm postępowania dla ustalonego kąta θ obrotu profilu odkształceń względem przekroju zespolonego wygląda następująco:
1. Przy ustalonym punkcie C tj. odkształceniu granicznym (ϵ_{tu}, y_{tu}) iterować wartością odkształcenia dla współrzędnej y_{cu} w zakresie $[\epsilon_{tu}, \epsilon_{cu}]$ (obrót profilu względem punktu C).
2. Przy ustalonym punkcie A tj. odkształceniu granicznym (ϵ_{cu}, y_{cu}) iterować wartością odkształcenia dla współrzędnej y_{tu} w zakresie $[\epsilon_{tu}, \epsilon_{co\_y\_tu}]$ (obrót profilu względem punktu

A). Gdzie $\epsilon_{co\_y\_tu}$ jest wartością odkształcenia wyznaczoną przez przecięcie prostej przechodzącej przez punkty A i B z prostą $y = y_{tu}$.

3. Przy ustalonym punkcie B tj. odkształceniu granicznym (ϵ_{co}, y_{co}) oraz początkowym kącie $\phi = \phi_{ini}$ dla profilu odkształceń wyznaczanego przez prostą przechodzącą przez punkty A i B iterować kątem ϕ w zakresie $[\phi_{ini}, 0]$ (obrót profilu względem punktu B). Zakłada się, że kąt ϕ jest dodatni gdy wywołuje ściskanie w górnych włóknach. Docelowo uzyskuje się profil odkształceń jak dla czystego ściskania tj. odkształcenia na całej wysokości równe tym w punkcie B.

7.3 Obrót profilu odkształceń względem przekroju zespolonego.

Algorytmy przedstawione zarówno w (Rozdział 7.2) jak i (Rozdział 7.3) umożliwiały wyznaczenie dopuszczalnych odkształceń granicznych $(\epsilon_{tu}, y_{tu}), (\epsilon_{cu}, y_{cu}), (\epsilon_{co}, y_{co})$, a następnie sterowanie profilem odkształceń w ich zakresie przy ustalonej wartości kąta θ . W celu uzyskania pełnego rozwiązania analizy nośności przekroju zespolonego w postaci krzywych i powierzchni interakcji należy zaprezentowane algorytmy umieścić wewnątrz pętli iterującej wartością kąta θ w zakresie $[0,360)$ stopni. Oznacza to sterowanie kierunkiem osi obojętnej profilu odkształceń wokół przekroju zespolonego. Dla każdego kierunku osi obojętnej należy wyznaczyć nowe wartości dopuszczalnych odkształceń granicznych $(\epsilon_{tu}, y_{tu}), (\epsilon_{cu}, y_{cu}), (\epsilon_{co}, y_{co})$ i przeprowadzić w ramach ich zakresu sterowanie profilem odkształceń. Sterowanie osią obojętną oraz profilem odkształceń zostało zaimplementowane w metodzie: `SectionSolver::solveSection(void)`.

Jako kroki iteracji przyjęto wartości Y_CU_STEP = Y_TU_STEP = 0.0001 oraz FI_STEP = 10^{-5} stopnia. Dla każdego wygenerowanego profilu odkształceń obliczane są wartości odkształceń odpowiadające y_{MIN}, y_{MAX} , a następnie na ich podstawie konstruowany jest obiekt `StrainProfile`, który jest przekazywany do metod całujących przekroju zespolonego. Wyniki całkowań dla poszczególnych profili zbierane są w kontenerze wierzchołków `QList<Vertex3D *> vertices`.

8. KONSTRUKCJA POWIERZCHNI I KRZYWYCH INTERAKCJI

Zgodnie z informacjami podanymi w części teoretycznej niniejszej pracy (Rozdział 2.4) do narysowania powierzchni oraz krzywych interakcji potrzebny jest zbiór punktów (M_x, My, N) w trójwymiarowej przestrzeni, które reprezentują wartości granicznej nośności przekroju zespolonego dla poszczególnych konfiguracji granicznego profilu odkształceń. Zbiór tych punktów otrzymuje się w wyniku całkowania naprężeń odpowiadających granicznym profilom odkształceń generowanym zgodnie z algorytmami podanymi w (Rozdział 7). Metoda `void SectionSolver::integrateStrainProfileFrom(double eps_y_tu, double y_tu, double eps_y_cu, double y_cu)` na podstawie wyznaczonych iteracyjnie (Rozdział 7.2) punktów $(\epsilon_{tu}, y_{tu}), (\epsilon_{cu}, y_{cu})$ konstruuje obiekty typu StrainProfile i przekazuje je do metod całkujących `Section::summateInternalActions_XXX(StrainProfile *p)`. Otrzymane siły przekrojowe są następnie transformowane to globalnego centralnego układu współrzędnych *XCY* (96),(97), gdzie *C* to ważony środek ciężkości przekroju. Otrzymane punkty (M_x, My, N) opakowywane są w obiekty typu `Vertex3D` i agregowane w kontenerze `QList<Vertex3D *>`.

8.1 Metody rysujące krzywe interakcji

W celu zobrazowania otrzymanego zbioru wierzchołków należy go uprzednio odpowiednio posortować. W klasie `SectionSolver` zdefiniowano cztery metody sortujące kontener `QList<Vertex3D *>` o nazwie `sortVerticesByXXX()`, gdzie za *XXX* można podstawić $N, M_x, M_y, Alfa$. Sortowanie wierzchołków względem *N* umożliwia narysowanie krzywych interakcji $M_x - M_y$, sortowanie względem M_x krzywych $M_y - N$, sortowanie względem M_y krzywych $M_x - N$ i w końcu sortowanie względem α krzywych $M - N$. *M* jest tutaj momentem wypadkowym momentów M_x, M_y , natomiast α to kąt jego nachylenia w

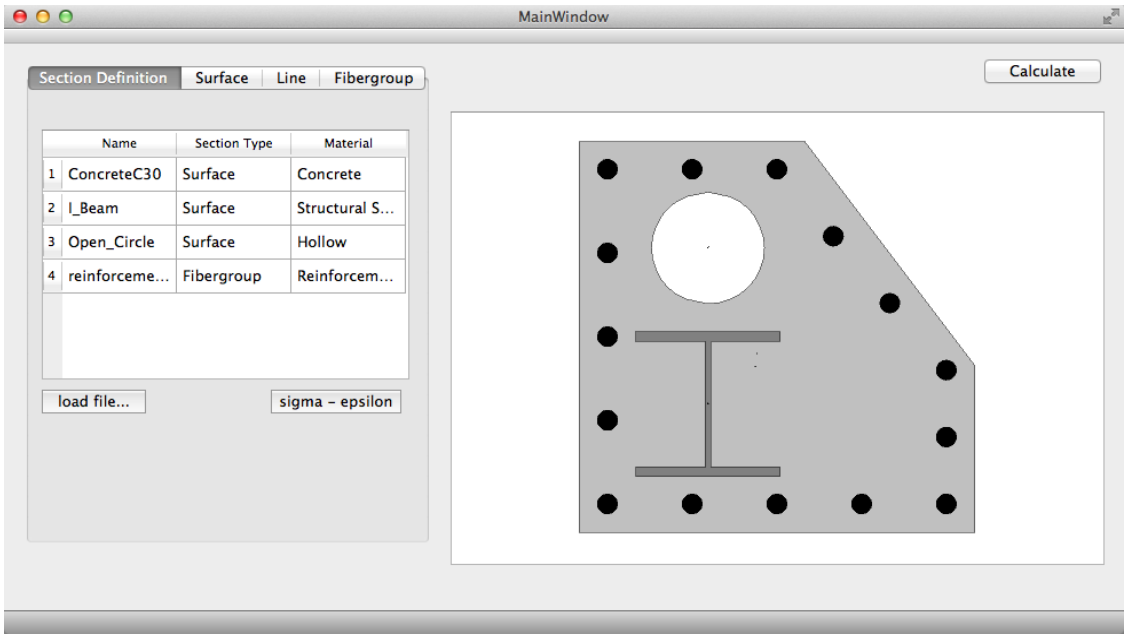
płaszczyźnie $M_x - M_y$ względem dodatniej osi OM_x. Metody sortujące wykorzystują wbudowaną funkcję `qSort()`, która jako ostatni argument pobiera obiekt funkcyjny definiujący sposób porównywania sortowanych elementów. Łącznie zdefiniowano osiem komparatorów `Vertex3DComparatorByXXX` w klasie `Vertex3D`. Program realizuje rysowanie wykresów krzywych interakcji w osobnym okienku `SolutionWindow` za pośrednictwem zdefiniowanych w tej klasie metod np. `SolutionWindow::drawInteractionCurveMxMyForN(int N)`. Metoda ta pobiera jako jedyny argument wartość siły osiowej N dla której ma zostać narysowana krzywa interakcji. Ze względu na stosunkowo dużą nierównomierność rozkładu punktów w przestrzeni w przypadku stosowania sterowania $\epsilon - \theta$ ($d - \theta$) (Rozdział 2.3) liczba otrzymywanych wierzchołków dla pojedynczej wartości siły osiowej N jest z reguły niewystarczająca. W celu rozwiązania tej niedogodności zaleca się zbierać punkty z przedziału $[N - X, N + X]$, gdzie zmienną X należy dobrać w zależności od potrzeb. Innym sposobem byłoby zastosowanie bardziej zaawansowanego algorytmu z użyciem przeszukiwania Brent'a (Rozdział 2.3). Posortowane i wybrane w ten sposób wierzchołki są następnie agregowane w obiekcie `QPolygonF` i rysowane z na `QGraphicsScene`. Aby zwiększyć czytelność wykresów w tle dodano układ współrzędnych oraz siatkę z zaznaczonymi wartościami, która skaluje się automatycznie względem rozmiaru krzywych interakcji. W tym celu rozszerzono klasę `QGraphicsScene` oraz zaimplementowano własną metodę `drawBackground()`.

8.2 Rysowanie powierzchni interakcji

Powierzchnie interakcji tworzą trójwymiarowe bryły stąd aby je narysować nie wystarczy użyć prostych kontrolek rysujących jak `QGraphicsScene`. Można by zastosować np. biblioteki OpenGL, aczkolwiek w przypadku realizowanego programu ograniczono się jedynie do eksportowania otrzymanego zbioru punktów do MathLab'a. Poniżej przedstawiono prostą procedurę skutecznie rysującą trójwymiarową powierzchnię interakcji w środkowisku MathLab.

9. PROGRAM KOMPUTEROWY

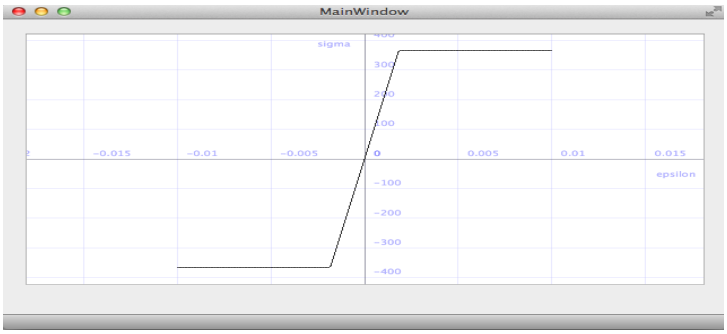
Przedstawiony program komputerowy zaimplementowano w całości w języku C/C++ wykorzystując środowisko programistyczne QT. Z założenia kod powinien być przenośny i kompilowalny na większości popularnych systemów operacyjnych, tj. Windows, Mac OS, Linux. Interfejs użytkownika bazuje na kontrolkach biblioteki QT (Rys. 38).



Rys. 38 Interfejs użytkownika, główne okno programu.

9.1 Okno główne aplikacji

Okno główne aplikacji zawiera po lewej stronie kontrolki umożliwiające definiowanie przekroju. Dostępne są cztery zakładki: SectionDefinition, Surface, Line i FiberGroup. Umożliwiają one odpowiednio przeglądanie wszystkich zdefiniowanych komponentów przekroju, definiowanie powierzchni, elementów liniowych oraz zbrojenia. Istnieją dwa zasadnicze sposoby wprowadzania danych wejściowych do programu. Pierwszym jest ręczne wpisanie wartości w odpowiednich zakładkach i kliknięcie przycisku „Save". Drugim wczytanie poszczególnych komponentów z pliku XML. Istnieje również możliwość załadowania całego przekroju z pliku XML. Po prawej stronie zdefiniowany przekrój jest natomiast wizualizowany. Efekt linearyzacji łuków widoczny jest dopiero po wykonaniu obliczeń przyciskiem „Calculate". Aplikacja umożliwia wybranie dowolnego z komponentów zdefiniowanych na liście, a następnie wizualizowanie przypisanego mu związku $\sigma - \epsilon$ za pomocą przycisku „Sigma-Epsilon" (Rys. 39). Ponadto poprzez dwukrotne kliknięcie na liście komponentów można przejść w tryb edycji wskazywanego komponentu.



Rys. 39 Wizualizacja związków konstytutywnych materiałów

9.2 Pliki XML z danymi wejściowymi

Dane wejściowe z definicją przekroju można wczytywać z plików XML. Strukturę przykładowego pliku przedstawiono poniżej (Rys. 40).

```
<?xml version="1.0" encoding="UTF-8"?>
<section>
<surface>
       <name>ConcreteC30</name>
       <material fck="30">Concrete</material>
       <opening>0</opening>
       <points>
        <point x="100" y="100" omega="0" />
        ...
       </points>
</surface>
<surface>
       <name>I_Beam</name>
       <material grade="S355">Structural Steel</material>
       <opening>0</opening>
       <points>
        <point x="200" y="200" omega="0" />
        …
       </points>
</surface>
<surface>
       …
</surface>
<fibergroup>
    <name>reinforcement_B500</name>
    <material grade="St3SYb500">Reinforcement Bars</material>
    <points>
            <point x="150" y="150" area="1018" />
             ….
    </points>
</fibergroup>
<line>
       <name>line_reinforcement</name>
       <material grade="B500SP">Reinforcement</material>
       <points>
        <point x="150" y="150" area="500" />
       ….
       </points>
</line>
</section>
```

Rys. 40 Struktura pliku XML z definicją przykładowego przekroju.

Warto zwrócić uwagę iż, aby wczytać komponent niezależnie np. tylko powierzchnię należy w pliku XML umieścić jako główny węzeł element <surface> (w przypadku powierzchni) lub <line> (w przypadku elementu liniowego) lub <fibergroup> (w przypadku zbrojenia). Struktura węzłów potomnych powinna być natomiast zgodna z tą w przedstawionym fragmencie pliku XML dla przekroju (Rys. 40).

9.3 Dane wyjściowe – wykresy krzywych interakcji

Dane wyjściowe aplikacji to rysowane w osobnym oknie programu wykresy krzywych interakcji (Rozdział 10).Istnieje możliwość wyboru jednej z czterech zakładek: $M_x - M_y$, $M_y - N$, $M_x - N$, $M - N$. W każdej z tych zakładek należy podać wartość parametru np. siły osiowej dla której chcemy narysować wykres krzywej interakcji. Dane wyjściowe możemy również wyeksportować do pliku jako listę punktów *Mx, My, N* przy czym każdy z nich umieszczony jest w nowej linii. Wyeksportowany plik można wykorzystać jako dane wejściowe do narysowania powierzchni interakcji w środowisku MathLab. (Rys. 8.2)

10. BENCHMARKI

Celem sprawdzenia poprawności działania zaprezentowanego algorytmu oraz programu przeprowadzonych zostanie kilka benchmarków.

10.1 Sprawdzenie poprawności całkowania numerycznego Gaussa-Legendre

Pierwszym testem będzie weryfikacja poprawności działania algorytmu całkowania numerycznego Gaussa. Jeżeli w całce powierzchniowej danej wzorem (98) przyjmiemy stałą wartość funkcji $\sigma(y)=1=const$ to wynikiem całkowania będą odpowiednio: (r,s) = (0,0) pole powierzchni, (r,s) = (0,1) moment statyczny S_x, (r,s) = (1,0) moment statyczny S_y, (r,s) = (0,2) moment bezwładności I_x, (r,s) = (2,0) moment bezwładności I_y, (r,s) =(1,1) moment dewiacji I_{xy}
Przyjęto przekrój betonowy 300x500mm, w klasie Concrete funkcja sigmaEpsilon() zwraca stałą wartość 1, odkształcenia graniczne betonu mają wartości: $\epsilon_{tu}=0, \epsilon_{co}=-0.002, \epsilon_{cu}=-0.0035$.
W funkcji `MainWindow::calculatePushButtonClicked()` umieszczono poniższy fragment kodu:

```
StrainProfile *profile = new StrainProfile(sectionSolver, M_PI*0/180, 0.0, -0.0035);
Surface *s = sectionSolver->getSurfaces()[0];
double area = s->internalAction_N(profile);
double S_x = s->internalAction_Mx(profile);
double S_y = s->internalAction_My(profile);
fprintf(stderr, "Area = %g, Sx = %g, Sy = %g\n", area, S_x, S_y);
```

Rys. 41 Fragment kodu liczący momenty statyczne

Otrzymane wartości: `Area = 150000, Sx = -7.45058e-09, Sy = -0`
Widzimy, że powierzchnia przekroju została poprawnie wyznaczona, a momenty statyczne względem o osi centralnych przekroju są w przybliżeniu równe 0. W kolejnym teście dla funkcji N przyjmiemy (r,s) = (0,2), dla M_x (r,s) = (2,0), dla M_y (r,s) =(1,1).

```
StrainProfile *profile = new StrainProfile(sectionSolver, M_PI*0/180, 0.0, -0.0035);
Surface *s = sectionSolver->getSurfaces()[0];
double I_x = s->internalAction_N(profile);
double I_y = s->internalAction_Mx(profile);
double I_xy = s->internalAction_My(profile);
fprintf(stderr, "Ix = %g, Iy = %g, Ixy = %g\n", I_x, I_y, I_xy);
```

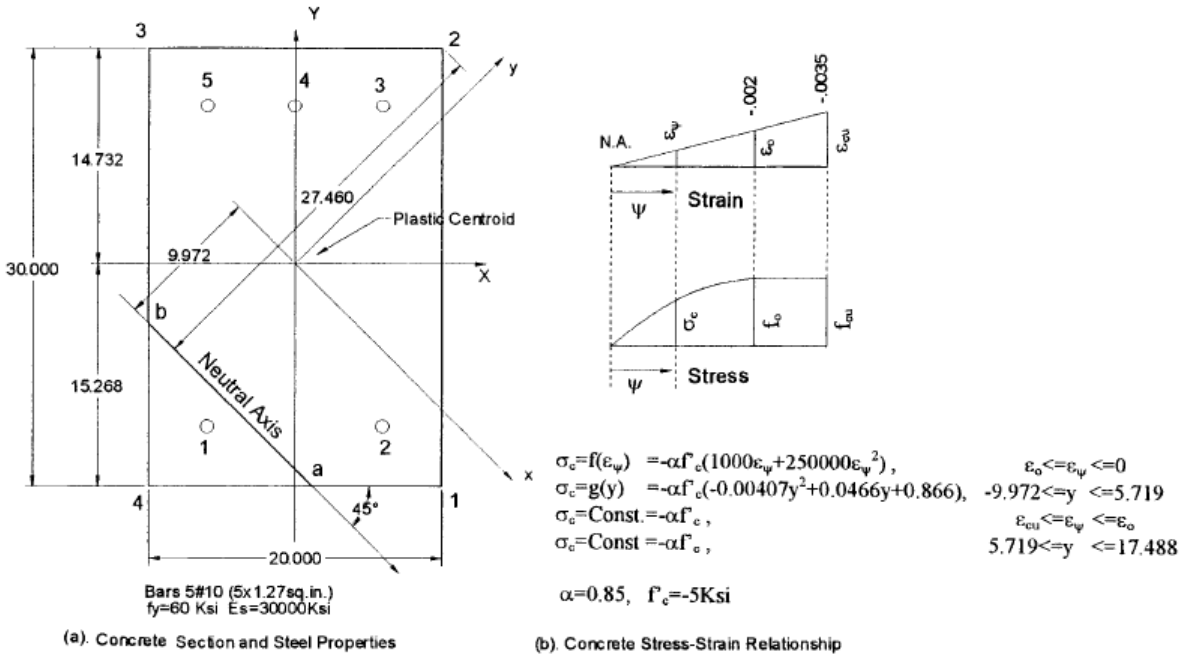
Rys. 42 Fragment kodu liczący momenty bezwładności i dewiacji

Otrzymane wartości: `Ix = 3.125e+09, Iy = 1.125e+09, Ixy = 5.96046e-08`
Wyniki obliczone ręcznie (Google): 300*500^3/12 = 3125000000 (dla Ix), 300^3*500/12 = 1125000000 (dla Iy), moment dewiacji prostokąta równy w przybliżeniu zero.

Dodatkowo przeprowadzono test sprawdzający momenty statyczne od górnej połowy przekroju, względem układu centralnego, przy kącie obrotu $\theta=90$ stopni. Fragment kodu jak na (Rys. 41) z tą różnicą, że profil odkształceń przyjmujemy następująco: `new StrainProfile (sectionSolver, M_PI*90/180, 0.0035, -0.0035)`.

Otrzymane wartości: `Area = 75000, Sx = 5.625e+06, Sy = 5.58794e-09`
Wyniki obliczone ręcznie (Google): 500*300/2 = 75000 (dla A), 500*(300/2)*(300/4) = 5625000 (dla Sx), lokalna oś y jest nadal osią centralną więc moment statyczny Sy w przybliżeniu jest równy zero.

Kolejny etapem weryfikacji poprawności algorytmu całkowania Gaussa będzie sprawdzenie wyników N, M_x, M_y otrzymanych programem z wynikami otrzymanymi przez Fafitis'a [19] dla przekroju żelbetowego (Rys. 43) przy kącie obrotu $\theta=-45$ i osi obojętnej $d=-9.972$ in.



Rys. 43 Sprawdzany przekrój żelbetowy z artykułu Fafitis'a [19]

Zależność naprężenie-odkształcenie dla betonu przyjęto jako paraboliczno-liniową (Rys. 43), natomiast dla stali jako sprężysto-plastyczną (model Prandtla) z granica plastyczności $f_{yk}=60\text{ksi}$ oraz modułem sprężystości $E_s=30000\text{ksi}$. Fragment kodu wykorzystany do przeprowadzenia testu przedstawiono na (Rys. 44).

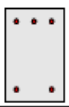
```
//FAFITIS EXAMPLE - STRAIN PROFILE TEST
double eps_low = (0.0035/27.460) * (y_MAX - y_MIN - 27.460);
double rotationAngle = (315.0/180.0)*M_PI;
StrainProfile *profile = new StrainProfile(sectionSolver, rotationAngle, eps_low, -0.0035);
double fi = profile->getCurvature_fi();
double d = profile->getNeutralAxis_d();
double e_o = profile->getStrainAtOrigin_eps_o();
double e_p = profile->getStrainAtLocalPoint(new Point(0,-9.972));
fprintf(stderr, "fi: %g, d: %g, e_o: %g, e_p: %g\n",
        fi, d, e_o, e_p);
double N = sectionSolver->summateInternalActions_N(profile);
double localeMx = sectionSolver->summateInternalActions_Mx(profile);
double localeMy = sectionSolver->summateInternalActions_My(profile);
double Mx = Section::transferBackFromLocalMx(localeMx, localeMy, rotationAngle); //[Nm]
double My = Section::transferBackFromLocalMy(localeMx, localeMy, rotationAngle); //[Nm]

fprintf(stderr, "N = %g\n", N);
fprintf(stderr, "locale Mx = %g\n", localeMx);
fprintf(stderr, "locale My = %g\n", localeMy);
fprintf(stderr, "Mx = %g\n", Mx);
fprintf(stderr, "My = %g\n", My);
```

Rys. 44 Fragment kodu testujący zgodność z obliczeniami Fafitis'a [19]

Wyniki działania powyższego kodu testowego przedstawiono na (Rys. 46), natomiast wartości odniesienia wyznaczone przez Fafitis'a i Papanikolau na (Rys. 45).

Comparison between Fafitis [17] and present integration scheme.

$\theta = -45°$, $d = -9.972$ in	N (kip)	M_x (kip-in)	M_y (kip-in)
Exact (100 Gauss points)	−1999.66	−7405.90	2820.60
Fafitis (reported values)	−1997.29	−7411.72	2796.55
Fafitis (simulated)	−1997.14	−7412.03	2797.00
Fafitis – error%	0.13%	0.08%	0.84%
Present strain-mapped scheme	−1999.66	−7405.90	2820.60



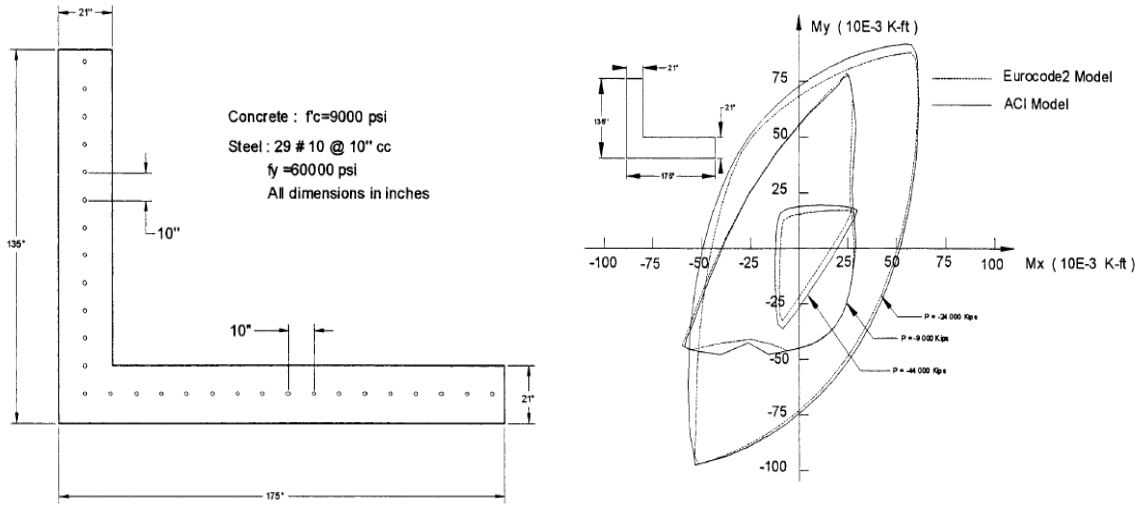
Rys. 45 Tablica z porównaniem wyników dla przekroju Fafitis'a [1]

```
Weighted A = 683.297, Sx = 10432.7, Sy = 6832.97
Geometrical center Cx: 10, Cy: 15.2682
fi: 0.000127458, d: -9.97197, e_o: -0.00127101, e_p: 3.88335e-09
N = -1999.65
locale Mx = -7231.24
locale My = -3242.31
Mx = -7405.92
My = 2820.6
```

Rys. 46 Wyniki działania kodu testowego dla przekroju żelbetowego z artykułu Fafitis'a [19]

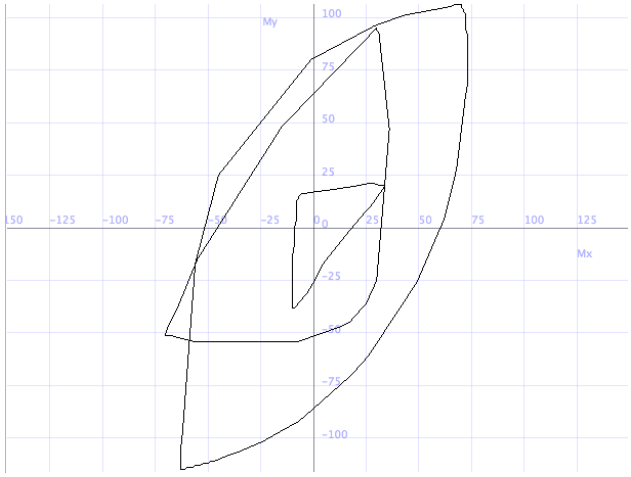
10. 2 Benchmarki przekrojów żelbetowych

Pierwszy przekrój żelbetowy, który zostanie przetestowany podchodzi z artykułu Fafitis'a [19] i został przedstawiony na (Rys. 47) wraz z otrzymanymi przez niego przykładowymi krzywymi interakcji.



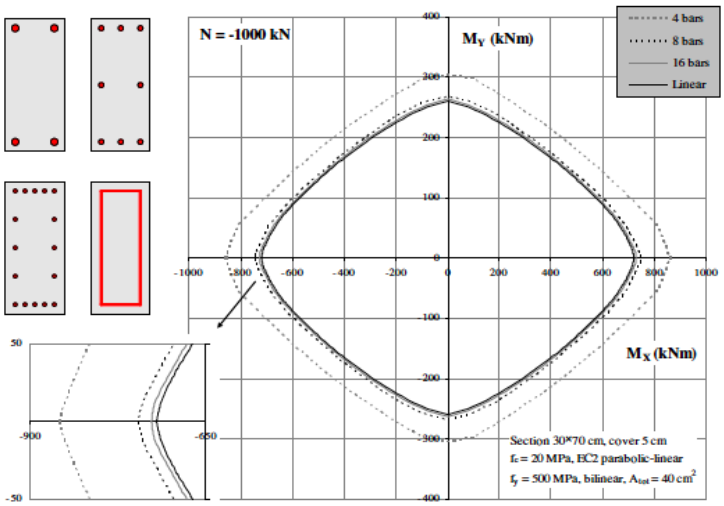
Rys. 47 Przekrój żelbetowy oraz krzywe interakcji z artykułu Fafitis'a [19]

Przyjęto następujące charakterystyki materiałowe: beton $f_c^{'}=9000\text{psi}$, stal $f_y=60000\text{psi}$, pozostałe dane jak w pierwszym testowym przykładzie od Fafitis'a. Wykresy otrzymane programem przedstawiono na (Rys. 48).

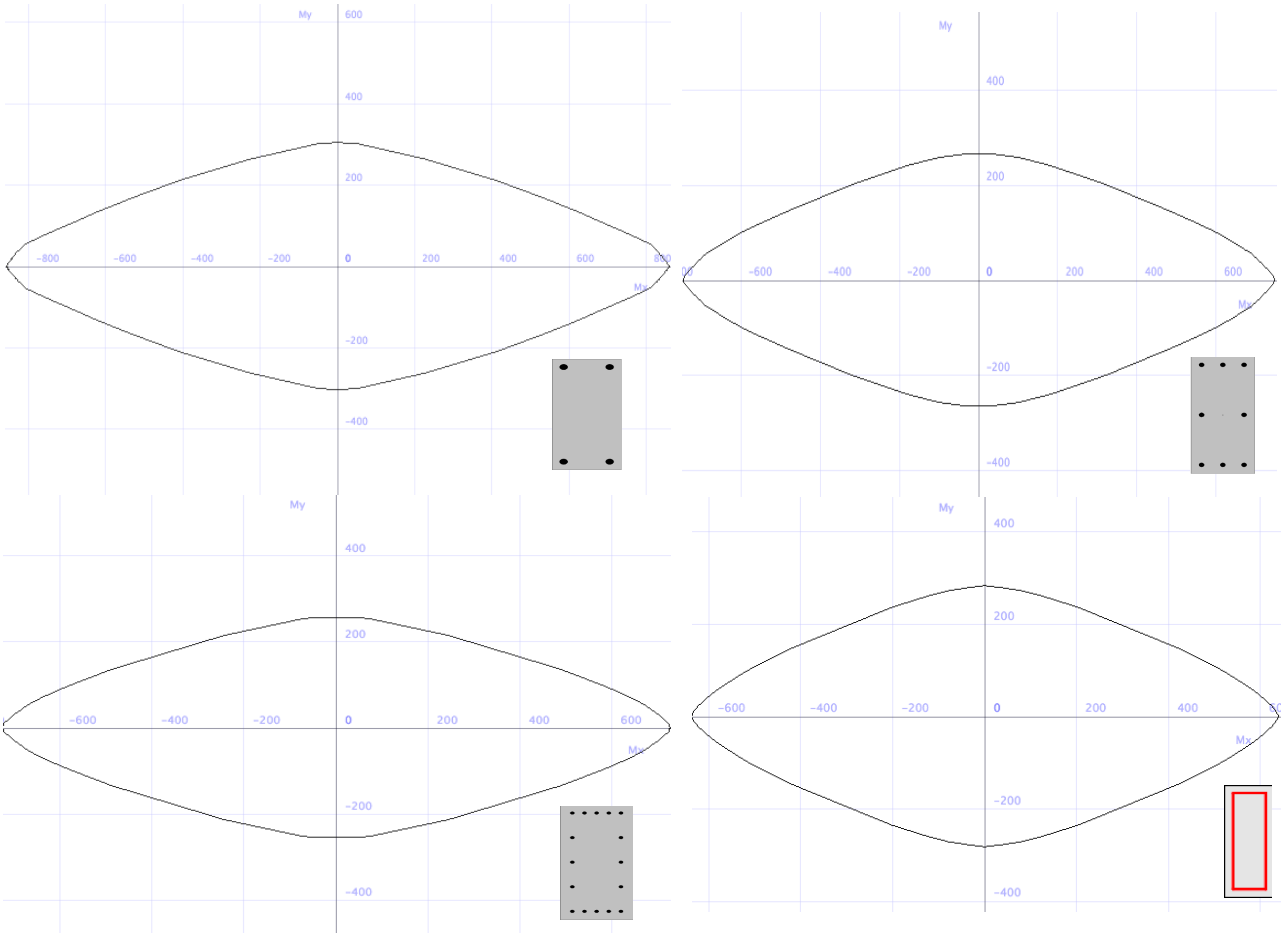


Rys. 48 Wykresy krzywych interakcji dla drugiego przekroju żelbetowego z Fafitisa'a

Kolejne testowane przekroje żelbetowe zaczerpnięto z artykułu Papanikolaou'a [1]. Poszczególne przekroje mają takie same wymiary komponentu betonowego 300x700mm, różnią się sposobem zdefiniowania zbrojenia (4, 8 lub 16 prętów zbrojeniowych oraz jako element liniowy), którego sumaryczna powierzchnia w każdym przypadku wynosi $A_{tot}=4000\text{mm}^2$. Dla betonu przyjęto wykres paraboliczno-liniowy odkształceń z EC2 oraz $f_{ck}=20\text{MPa}$. Natomiast dla zbrojenia przyjęto model Prandtla (sprężysto-plastyczny) z $\epsilon_{uk}=2\%$ oraz granicą plastyczności $f_{yk}=500\text{MPa}$. Wzorcowe wykresy przedstawiono na (Rys. 49), obliczone na (Rys. 50)

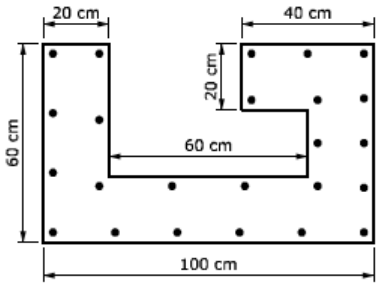


Rys. 49 Wzorcowy wykresy krzywych interakcji dla przekrojów żelbetowych [1]



Rys. 50 Wykresy krzywych interakcji obliczone programem

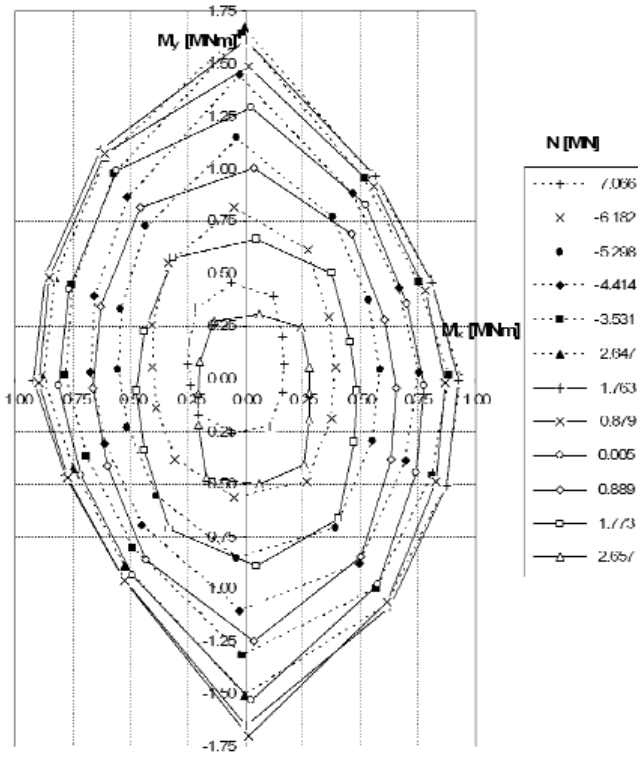
Tym razem przeprowadzony zostanie benchmark dla przekroju żelbetowego o niesymetrycznym kształcie. Jako wzorcowy wybrano przekrój w kształcie litery G (ang. G-shaped section) zaproponowany przez Rosati'ego [22]. Geometrię przekroju oraz charakterystyki materiałowe przedstawiono na (Rys. 51), wykresy porównawcze na (Rys. 52), natomiast krzywe narysowane programem na (Rys. 53).



Concrete:
$f_{ck} = 20.75\, N/mm^2$
$\gamma_c = 1.6$

Reinforcement bars:
$f_{yk} = 375\, N/mm^2$
$\gamma_b = 1.15$
$\phi 24$ at the corners
$\phi 24$ on the faces
(equally spaced between the bars at the corners)
concrete cover: $3cm$

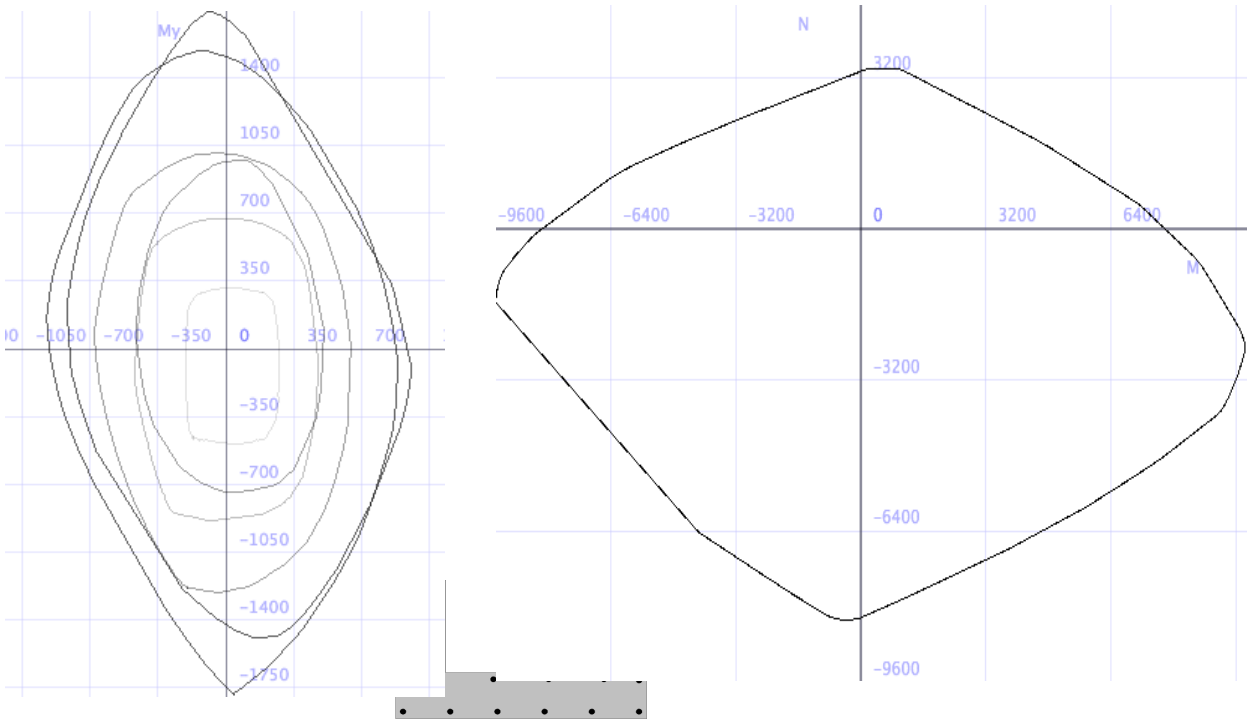
Rys. 51 Geometria i charakterystyki materiałowe przekroju od Rosati'ego [22]



$$\mathbf{f}_{N_{\max}} = \{3.5404,\ 0.0738,\ -0.0811\}$$
$$\mathbf{f}_{N_{\min}} = \{-7.9498,\ -0.0738,\ 0.0811\}$$

Rys. 52 Krzywe interakcji dla przekroju w kształcie litery G, Rosati [22]

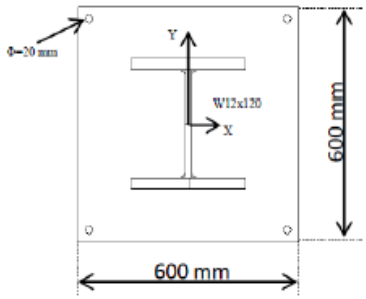
Wygenerowano krzywe dla następujący wartości siły osiowej -6182 kN, -3531kN, -879kN, 889kN, 1773kN, 2657kN. Na (Rys. 53) po prawej stronie przedstawiono również przykładową krzywą interakcji M-N dla kąta $\alpha = 0$.



Rys. 53 Krzywe interakcji Mx-My (po lewej) oraz M-N (po prawej) dla przekroju Rosati'ego.

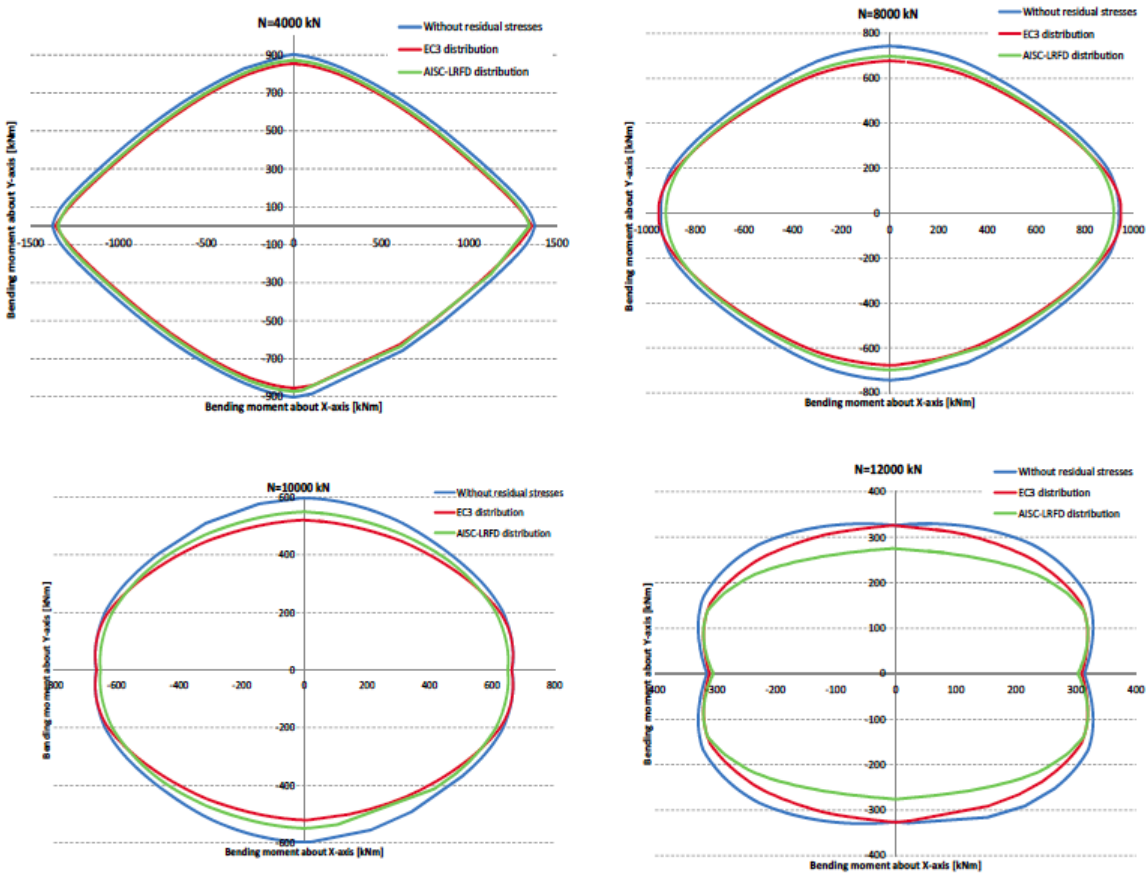
10. 3 Benchmarki dla przekrojów zespolonych

W celu weryfikacji poprawności programu dla przekrojów zespolonych jako pierwszy przypadek testowy wybrano przekrój zaproponowany przez C.G. Chiorean [20] (Rys. 51)



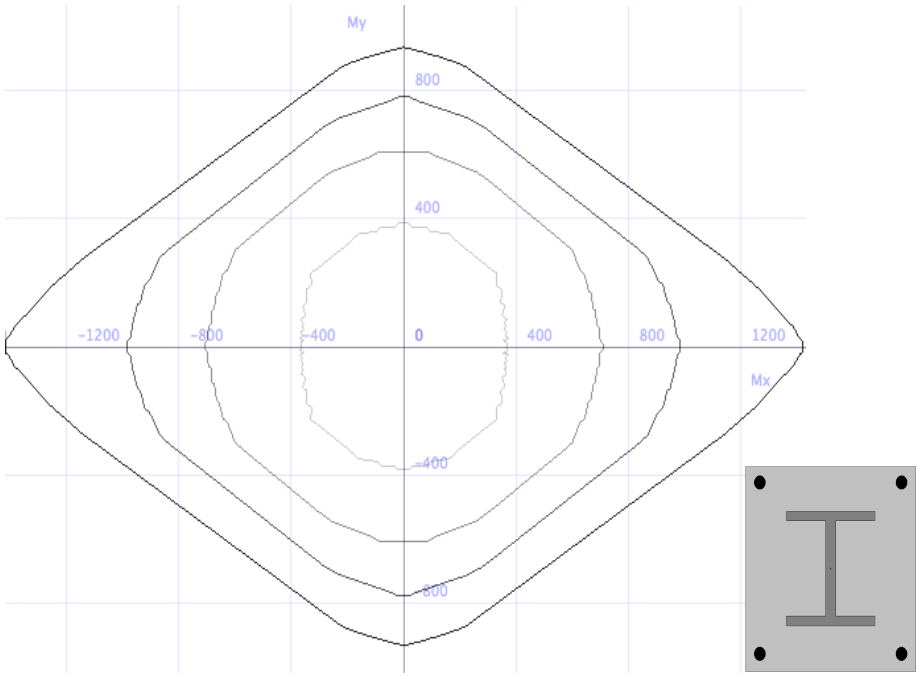
Rys. 51 Przekrój zaproponowany przez Chiorean'a [20]

Na (Rys. 52) przedstawiono wzorcowe wykresy dla $N = [-4000, -8000, -10000, -12000]\, kN$



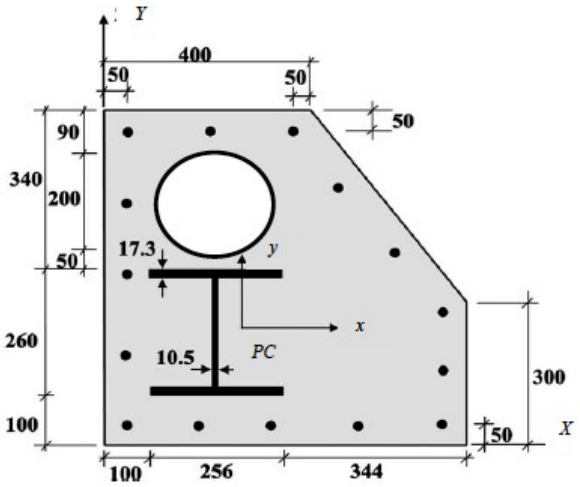
Rys. 52 Wzorcowe wykresy krzywych interakcji przekroju zespolonego Chiorean [20]

Natomiast na (Rys. 53) odpowiadające im wykresy wygenerowane programem. Przyjęto wytrzymałość charakterystyczną betonu na ściskanie $f_{ck}=20\text{MPa}$, wykres $\sigma-\epsilon$ paraboliczno-liniowy. Dla stali przyjęto granicę plastyczności $f_{yk}=300\text{MPa}$ i moduł sprężystości $E_s=200\text{GPa}$. Graniczne odkształcenia w modelu Prandtla wynoszą $\epsilon_{uk}=0.01$.

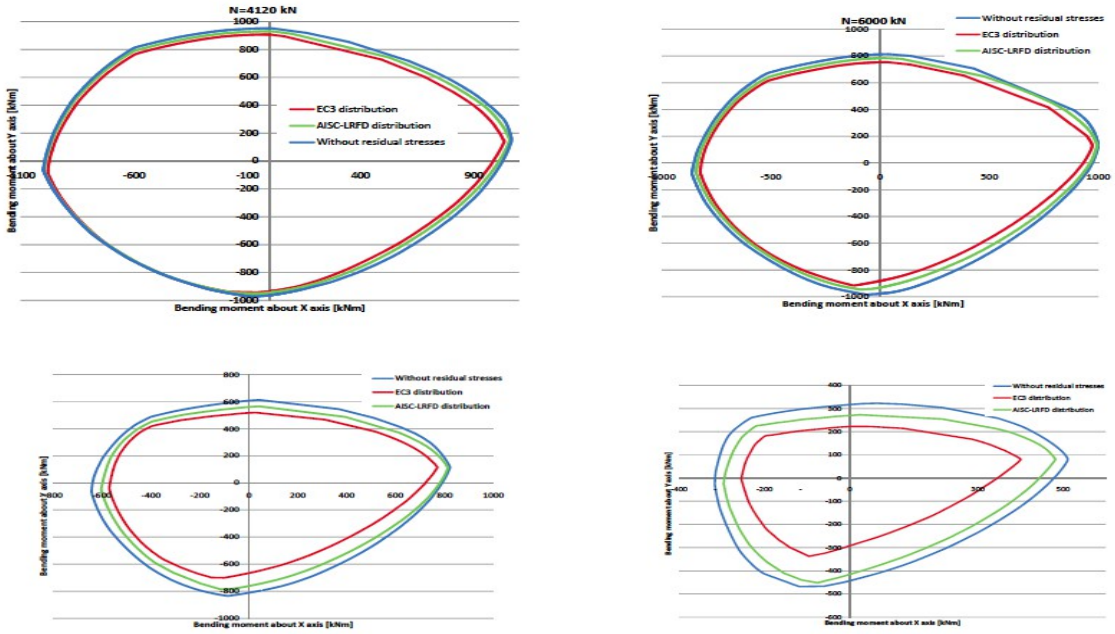


Rys. 53 Wykresy krzywych interakcji z programu dla przekroju Chiorean'a

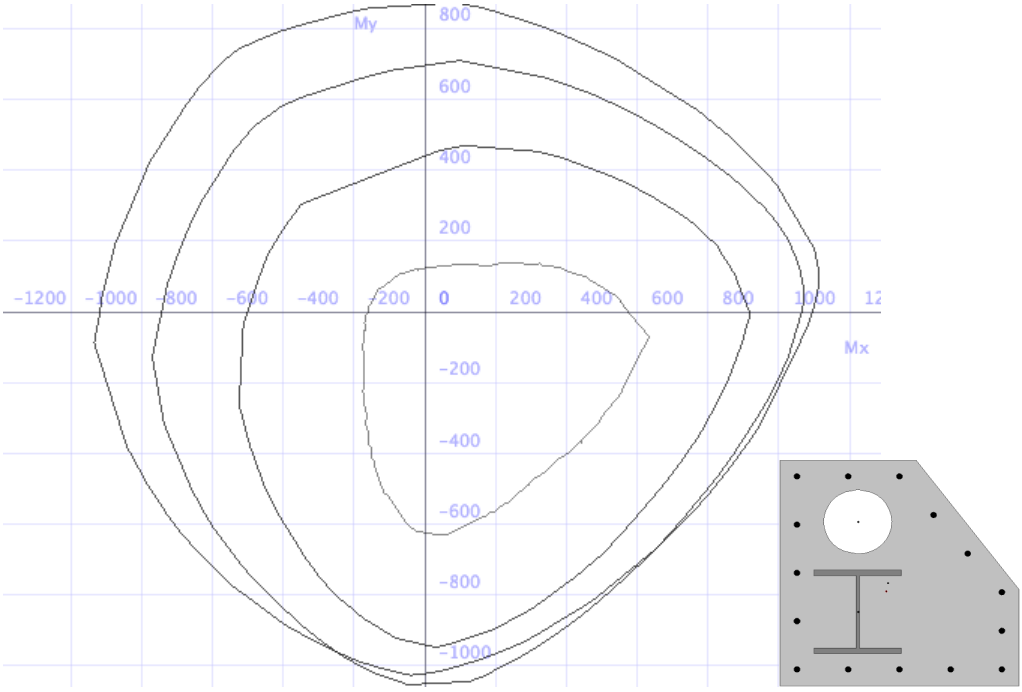
Kolejny testowy przekrój zespolony pochodzi od Chen et al. [21], był on powszechnie wykorzystywany jako benchmark zarówno przez Papanikolao'a [1], Chiorean'a [20] jak i Rosati'ego [22]. Geometrię przekroju przedstawiono na (Rys. 54). Charakterystyki materiałowe przyjęto zgodnie z artykułem Chiorean'a [20] tj. charakterystyczna wytrzymałość betonu $f_{ck}=30\text{MPa}$, granica plastyczności stali dwuteownika $f_{st}=355\text{MPa}$, granica plastyczności stali zbrojeniowej $f_s=460\text{MPa}$. Podane wartości zostały zredukowane częściowymi współczynnikami bezpieczeństwa $\gamma_c=1.50, \gamma_{st}=1.10, \gamma_s=1.15$. Dla betonu przyjęto model paraboliczno-liniowy zależności $\sigma-\epsilon$ z wartościami granicznych odkształceń $\epsilon_{co}=0.002, \epsilon_{cu}=0.0035$. Dla stali w obu przypadkach model sprężysto-plastyczny z granicznymi odkształceniami $\epsilon_{uk}=0.1$. Wzorcowe wykresy krzywych interakcji dla $N=[-4120,-6000,-8000,-1000]kN$ przedstawiono na (Rys. 55), natomiast wykresy wygenerowane programem na (Rys. 56).



Rys. 54 Geometria przekroju testowego od Chen et al. [21] na podstawie Chiorean'a [20]



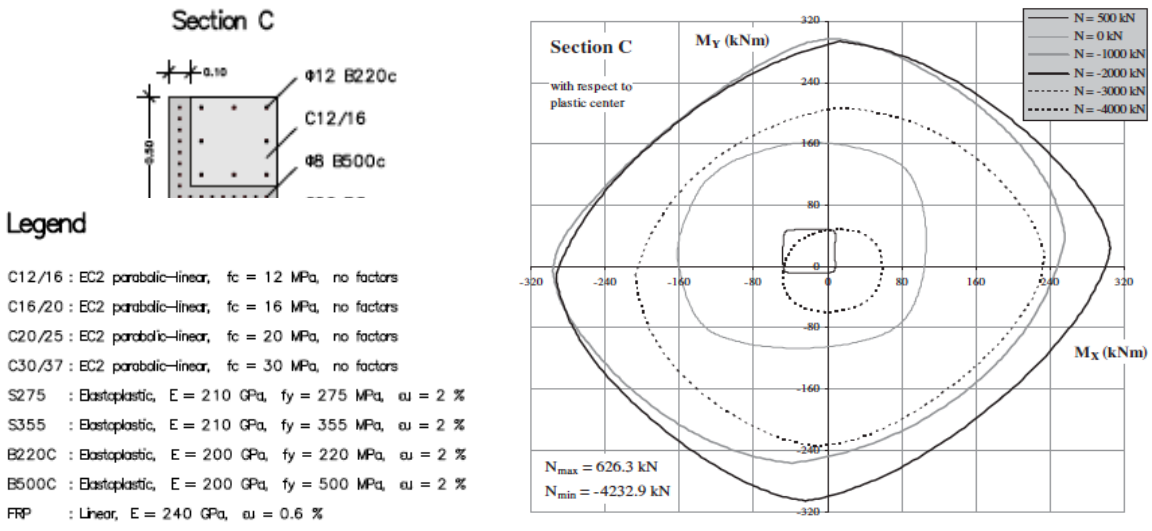
Rys. 55 Wzorcowe wykresy dla przekroju od Chen et al. [21][20]



Rys. 56 Wykresy krzywych interakcji otrzymane programem dla przekroju Chen et al.

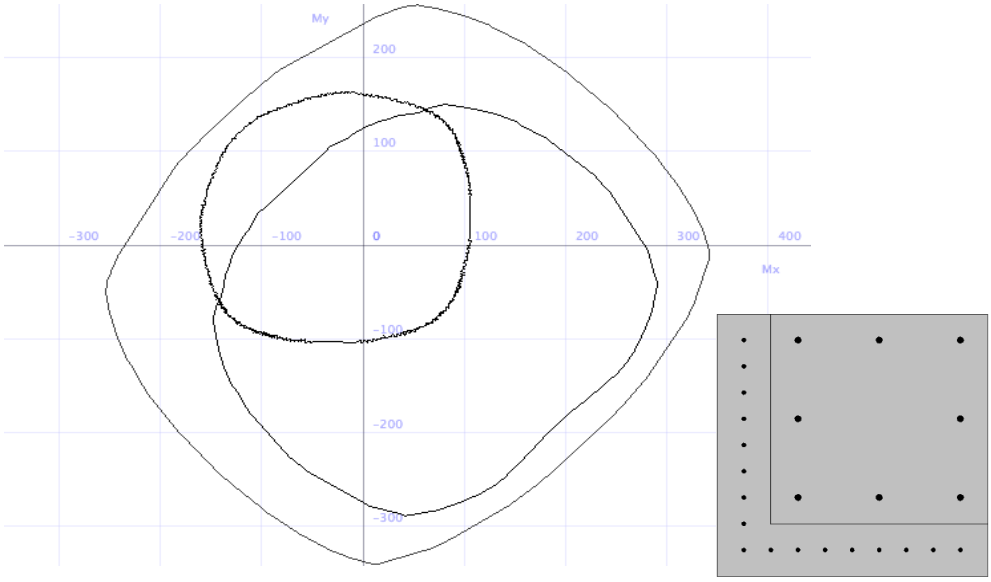
Na wykresach przedstawionych na (Rys. 56) można zauważyć nieznaczne rozbieżności w stosunku do (Rys. 55), może to wynikać z zastosowania przez Chiroean'a [20] innej metody całkowania naprężeń (Rys. 45 – porównanie różnych metod) oraz z nieco niedokładnego wyznaczenia środka ciężkości bezpośrednio zależnego od linearyzacji otworu kołowego jak również rozmieszczenia zbrojenia na ukośnej krawędzi przekroju (trudność dokładnego określenia jego położenia).

Ciekawym przypadkiem do przetestowania jest przekrój zespolony (Section C) zaproponowany przez Papanikolaou'a [1]. Geometrię przekroju przedstawia (Rys. 57), jest on zespolony z dwóch części z których każda wykonana jest z betonu innej klasy. W omawianym przekroju zastosowano również dwa różne gatunki stali zbrojeniowej. Na (Rys. 57) po prawej stronie pokazano również wzorcowe wykresy krzywych interakcji z artykułu [1].



Rys. 57 Przekrój zespolony Section C od Papanikoloau'a [1]

Wykresy krzywych interakcji $M_x - M_y$ dla siły osiowej N = -3000kN, -2000kN, 0 kN przedstawiono na (Rys. 58) .



Rys. 58 Krzywe interakcji wygenerowane programem dla przekroju Section C, Papanikolaou [1]

BIBLIOGRAFIA

[1] Vassilis K. Papanikolaou, Analysis of arbitrary composite section in biaxial bending and axial load
[2] Antoni Biegus, Projektowanie zespolonych konstrukcji stalowo-betonowych wg Eurokodu 4
[3] PN-EN 1994-1-1, Eurokod 4: Projektowanie zespolonych konstrukcji stalowo-betonowych – Część 1-1: Reguły ogólne i reguły dla budynków.
[4] PN-EN 1992-1-1, Eurokod 2 - Projektowanie konstrukcji z betonu - Część 1-1: Reguły ogólne i reguły dla budynków
[5] PN-EN 1993-1-1, Eurokod 3: Projektowanie konstrukcji stalowych - Część 1-1: Reguły ogólne i reguły dla budynków
[6] Janusz German, Wykłady z wytrzymałości materiałów dla studiów stacjonarnych, 15. Analiza stanu naprężenia w belkach zespolonych
[7] Adam P. Zaborski, Krzywe interakcji M-N w stanie nośności granicznej sprężystej i plastycznej przekroju
[8] Adam P. Zaborski, Belki złożone i zespolone – wymiarowanie połączeń, przekrój żelbetowy, belki zespolone
[9] PN-EN 1990, Eurokod - Podstawy projektowania konstrukcji
[10] Andrzej Łapko, Builder, ZESZYT 2 - Projektowanie konstrukcji żelbetowych
[11] http://pl.wikipedia.org/wiki/Statyczna_próba_rozciągania
[12] http://en.wikipedia.org/wiki/Numerical_integration
[13] http://en.wikipedia.org/wiki/Rectangle_method
[14] http://en.wikipedia.org/wiki/Gaussian_quadrature
[15] http://en.wikipedia.org/wiki/Legendre_polynomials
[16] ACI, Building Code Requirements for Structural Concrete and Commentary 318-08. Detroit: American Concrete Institute.
[17] http://mathworld.wolfram.com/GreensTheorem.html
[18] http://mathworld.wolfram.com/PolygonArea.html
[19] Fafitis A., Fellow, ASCE, Interaction Surfaces of Reinforced-Concrete Sections in Biaxial Bending. J Struct Eng, ASCE 2001; 127:840-6.
[20] Cosmin G. Chiorean, A Computer Method for Rapid Design Of Composite Steel-concrete Cross-sections
[21] Chen SF, Teng JG, Chan SL, Design of biaxially loaded short composite columns of arbitrary section, J Struct Eng, ASCE 2001; 127:840-6.
[22] Rosati L, Marmo F, Serpieri R. Enhanced solution strategies for ultimate strength analysis of composite steel-concrete sections subject to axial force and biaxial bending, Comput Methods Appl Mech Eng 2008; 197; 1033-55