

**Jacek Chróścielewski
Mikołaj Miśkiewicz
Łukasz Pyrzowski**

WPROWADZENIE DO MODELOWANIA MES W PROGRAMIE SOFiSTiK

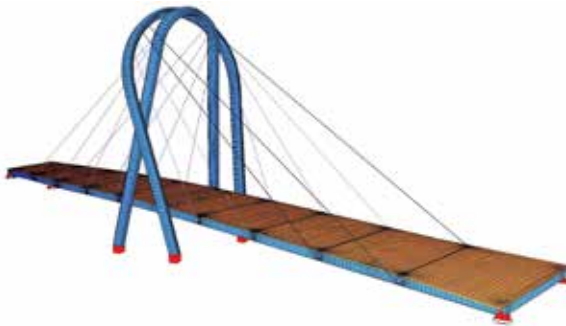
**Materiały pomocnicze
do laboratorium
z metody elementów
skończonych**

Wydawnictwo Politechniki Gdańskiej

Jacek Chróścielewski, Mikołaj Miśkiewicz, Łukasz Pyrzowski

Wprowadzenie do modelowania MES w programie SOFiSTiK

**Materiały pomocnicze do laboratorium
z metody elementów skończonych**



Gdańsk 2016

PRZEWODNICZĄCY KOMITETU REDAKCYJNEGO
WYDAWNICTWA POLITECHNIKI GDAŃSKIEJ
Janusz T. Cieśliński

RECENZENT
Wojciech Witkowski

REDAKCJA JĘZYKOWA
Agnieszka Frankiewicz

PROJEKT OKŁADKI
Wioleta Lipska-Kamińska

Na okładce przedstawiono wizualizacje modeli MES wykonanych w środowisku SOFiSTiK.
Od góry – kładka dla pieszych w Ustce, Forum Gdańsk, kładka dla pieszych w Radomiu,
most kolejowy przez rz. Martwa Wisła w Gdańsku, most przez rz. Brda w Bydgoszczy,
estakada na terenie Stoczni Gdańsk, most przez rz. Wisłok w Rzeszowie.

Wydano za zgodą
Rektora Politechniki Gdańskiej

Oferta wydawnicza Politechniki Gdańskiej jest dostępna pod adresem
<http://www.pg.edu.pl/wydawnictwo/katalog>
zamówienia prosimy kierować na adres wydaw@pg.gda.pl

Utwór nie może być powielany i rozpowszechniany, w jakiegokolwiek formie
i w jakikolwiek sposób, bez pisemnej zgody wydawcy

© Copyright by Wydawnictwo Politechniki Gdańskiej, Gdańsk 2016

ISBN 978-83-7348-670-6

WYDAWNICTWO POLITECHNIKI GDAŃSKIEJ

Wydanie I. Ark. wyd. 6,3, ark. druku 6,75, 1131/934

Druk i oprawa: Totem.com.pl, sp. z o.o., sp. k.
ul. Jacewska 89, 88-100 Inowrocław, tel. 52 354 00 40

SPIS TREŚCI

1. PRZEDMOWA	5
2. INFORMACJE OGÓLNE	7
2.1. Struktura programu	7
2.2. Praca w programie	7
3. UKŁADY PRĘTOWE	11
3.1. Materiały i przekroje	11
3.2. Definiowanie geometrii – SOFIMSHA	18
3.3. Obciążenia, obliczenia statyczne	26
3.4. Prezentacja wyników – atrybuty modelu obliczeniowego, przemieszczenia, siły	33
3.5. Prezentacja wyników – naprężenia	38
3.6. Wykorzystanie zmiennych oraz pętli – obciążenia ruchome	40
3.7. Kombinacje obciążeń i obwiednie	43
3.8. Uwagi o zbieżności rozwiązań MES, udoskonalenie typu h	47
4. UKŁADY POWIERZCHNIOWE	51
4.1. Materiały i definiowanie geometrii – SOFIMSHA	51
4.2. Obciążenia, obliczenia statyczne	58
4.3. Prezentacja wyników	60
4.4. Uwagi o zbieżności rozwiązań MES, udoskonalenie typu h	63
5. UKŁADY POWIERZCHNIOWO-PRĘTOWE	66
5.1. Materiały, przekroje, geometria – SOFIMSHA	66
5.2. Definiowanie geometrii – SOFIMSHC	72
5.3. Materiały, przekroje, geometria – SOFIPLUS	80
5.4. Prezentacja wyników – siły wewnętrzne	93
5.5. Prezentacja wyników – naprężenia	95
6. ZADANIA DODATKOWE – ROZWIĄZANIA	100
7. BIBLIOGRAFIA	105

1. PRZEDMOWA

Trudno obecnie znaleźć obszar działalności inżynierskiej powiązanej z obliczeniami konstrukcji, w którym nie byłaby wykorzystywana metoda elementów skończonych (MES). Powszechność zastosowań MES, tworzenie coraz bardziej przyjaznych użytkownikom, zautomatyzowanych środowisk systemów obliczeniowych mogą sugerować sposób ich wykorzystywania na zasadzie odpowiadającej prostym działaniom na kalkulatorze. Ta łatwość uzyskania wyników oraz ich atrakcyjna wizualizacja mogą prowadzić do mylnego przekonania, że jeśli obliczenia zakończyły się sukcesem, to muszą one być poprawne. Niestety tak nie jest. Efektywne i poprawne wykorzystanie komercyjnych systemów obliczeniowych poza znajomością ich obsługi (tzw. klikania) wymaga zrozumienia stosowanych w nich metod, ich podstaw matematycznych, w tym przyjętych teoretycznych założeń i hipotez, a co się z tym wiąże – ograniczeń z nich wynikających, oraz gruntownej wiedzy na temat badanych zjawisk fizycznych. Dlatego uzyskiwane wyniki obliczeń zawsze powinny być możliwie najdokładniej weryfikowane, co najmniej pod kątem jakościowym, np. w formie uproszczonych oszacowań analitycznych, analizy jednostek, kontroli symetrii, bilansu reakcji, zgodności z warunkami brzegowymi czy zachowania ciągłości pól wyznaczanych zmiennych stanu.

W tym miejscu należy mocno zaakcentować istotę i wagę wspomnianych wcześniej problemów, w tym opracowaniu bowiem nie zostanie im poświęcona znacząca uwaga. Zakres niniejszej publikacji jest ograniczony do technicznej strony użytkowania pre- i postprocesora systemu SOFiSTiK (tzw. klikania) w odniesieniu do zadań statyki liniowej. Należy jednak podkreślić, że cele opracowania nie obejmują jedynie prezentacji możliwości programu i nauki jego wykorzystania, a przede wszystkim zobrazowanie głównych zasad obowiązujących w metodzie elementów skończonych.

System SOFiSTiK należy do grupy programów wyspecjalizowanych, ukierunkowanych na zastosowania głównie w obszarze budownictwa (specjalistyczna baza norm). Program SOFiSTiK ma stosunkowo ubogą bibliotekę elementów skończonych, trzeba jednak podkreślić, że są one starannie wyselekcjonowane pod względem efektywności i niezawodności w odniesieniu do typowych układów konstrukcyjnych spotykanych w budownictwie. Takie podejście twórców programu – brak możliwości wyboru, a co się z tym wiąże konieczność samodzielnej oceny trafności użycia określonego elementu skończonego do rozwiązania konkretnego problemu – pozornie nie stawia przed jego użytkownikami wygórowanych wymagań dotyczących wiedzy o metodzie elementów skończonych. Pozornie, ponieważ nawet pracując z programem z tej grupy wyspecjalizowanych systemów, użytkownik powinien mieć opanowaną wiedzę z zakresu technik dyskretyzacyjnych oraz – i przede wszystkim – świadomość ograniczeń zakresu stosowalności wykorzystywanych elementów skończonych. Niestety w tej grupie programów w instrukcjach użytkownika informacja o podstawach teoretycznych jest najczęściej ograniczona do minimum lub całkowicie pominięta, często tylko z ubogim odwołaniem do literatury źródłowej. Dlatego niezbędnym minimum, poza

zapoznaniem się ze stosowną literaturą, powinno być osobiste przetestowanie dostępnych elementów skończonych przed przystąpieniem do profesjonalnego wykorzystania programu do obliczeń konstrukcji, tak aby „nabrać zaufania” do nich i rozpoznać ich ewentualne mankamenty. Temu celowi służą tzw. zadania benchmarkowe dostępne w literaturze czy nawet opracowane i zawarte w instrukcjach do programu. Celowe i bardzo nośne poznawczo jest sporządzenie własnych testów w postaci prostych akademickich zadań opartych na wiedzy nabytej podczas kursów wytrzymałości materiałów, mechaniki budowli czy teorii sprężystości, geotechniki i fundamentowania oraz dynamiki budowli. Wskazane jest wariantowe powtórzenie na drodze numerycznej poznanych i omawianych tam rozwiązań ram, tarcz, płyt, powłok, fundamentów, dla których znane są niekwestionowane rozwiązania analityczne (zob. np. RADWAŃSKA [2013]).

Nauka za pomocą skryptu metody elementów skończonych w środowisku programu SOFiSTiK w wersji 2016 ma charakter sekwencyjny; aby korzystać z dalszego materiału, należy się zapoznać z materiałem poprzedzającym.


Dla przybliżenia podstaw teoretycznych oraz praktycznego zastosowania MES autorzy proponują następujące opracowania: HUGHES [2000], MACNEAL [1978], MACNEAL I IN. [1998], MIŚKIEWICZ [2016], PYRZOWSKI [2011], RAKOWSKI I KACPRZYK [2005], WILSON [2002], ZIENKIEWICZ [1972] oraz ZIENKIEWICZ I TAYLOR [2005].

Niniejsza publikacja bazuje na materiałach dydaktycznych udostępnianych studentom w ramach kursu laboratoryjnego *metoda elementów skończonych* (CHRÓŚCIELEWSKI I IN. [2010]). Autorzy dołożyli wielu starań, by przedstawiony materiał był wolny od błędów. Jednak w przypadku wystąpienia jakichkolwiek pomyłek będziemy wdzięczni za wszelkie informacje.

2. INFORMACJE OGÓLNE


2.1. Struktura programu

Program SOFiSTiK, a precyzyjniej – środowisko systemu SOFiSTiK, charakteryzuje budowa modułowa. SOFiSTiK składa się z zestawu programów/ modułów (tabl. 2.1), pomiędzy którymi wymiana informacji następuje poprzez wspólną bazę danych (CDBase). Tak jak w większości programów MES, również tu można wyszczególnić moduły odpowiedzialne za przygotowanie danych (PREPROCESSING), wykonanie obliczeń (PROCESSING) oraz prezentację i dalszą obróbkę uzyskanych wyników (POSTPROCESSING).

Dodatkowo występuje tu możliwość importowania i eksportowania danych z innych środowisk, które wymieniono w kolumnie DATA-EXCHANGE. Z poszczególnymi modułami można pracować jak z niezależnymi programami lub skorzystać z platformy *Structural Desktop*  (SSD) integrującej system SOFiSTiK, zapewniającej bezpośrednią interakcję między poszczególnymi modułami.

2.2. Praca w programie

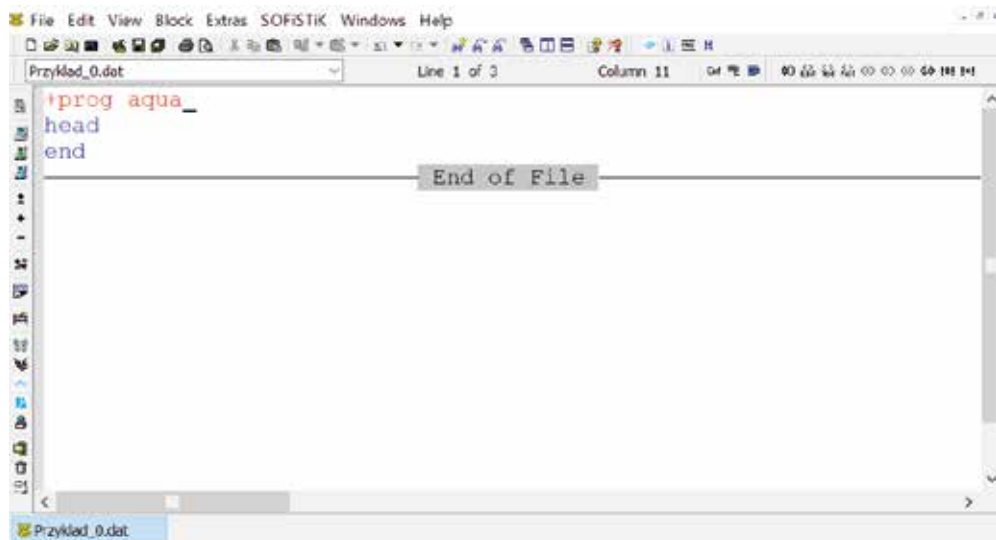
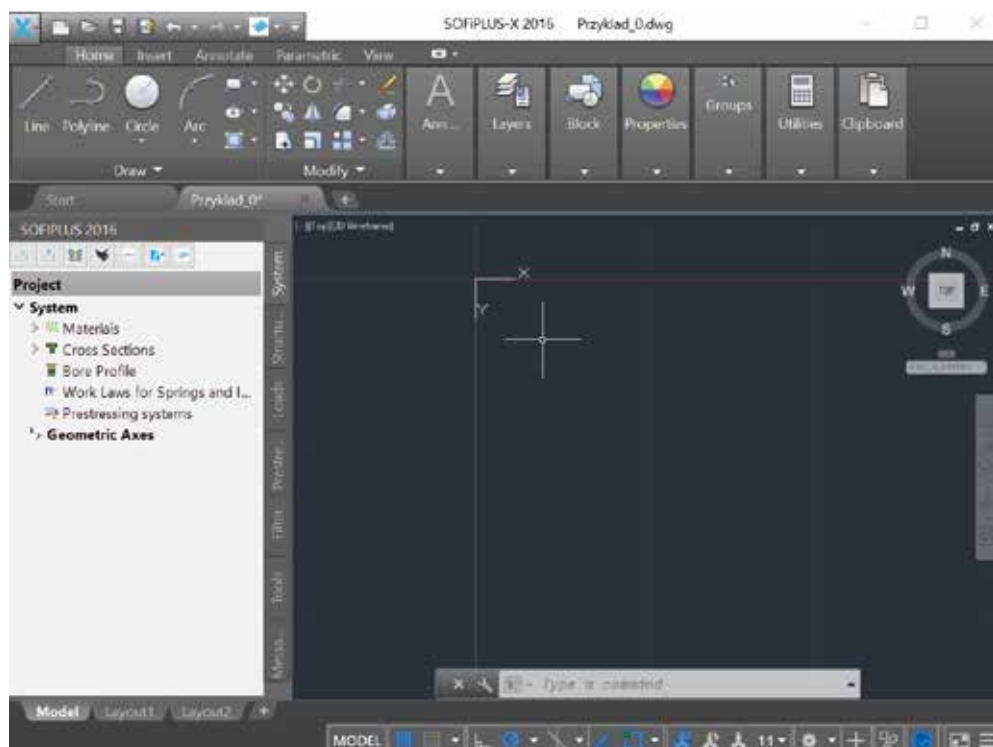
Organizacja pracy w programie jest ściśle związana z kolumnami w tabeli 2.1. W pierwszej kolejności użytkownik wybiera moduł, w którym wprowadza dane, a następnie ten, w którym będzie wykonywał obliczenia w zależności od rozwiązywanego problemu. Ostatecznie wskazuje, w jaki sposób chce zaprezentować wyniki obliczeń (dane tekstowe, wykresy, wizualizacja deformacji konstrukcji). Praca w SOFiSTiK pozwala użytkownikowi na dość swobodny wybór środowiska, w którym będzie wprowadzał dane dotyczące konstrukcji i definiował działające na nią obciążenia. Wykonanie obliczeń i obróbka wyników każdorazowo przebiegają w ten sam sposób, co zostanie wyjaśnione w dalszej części opracowania.

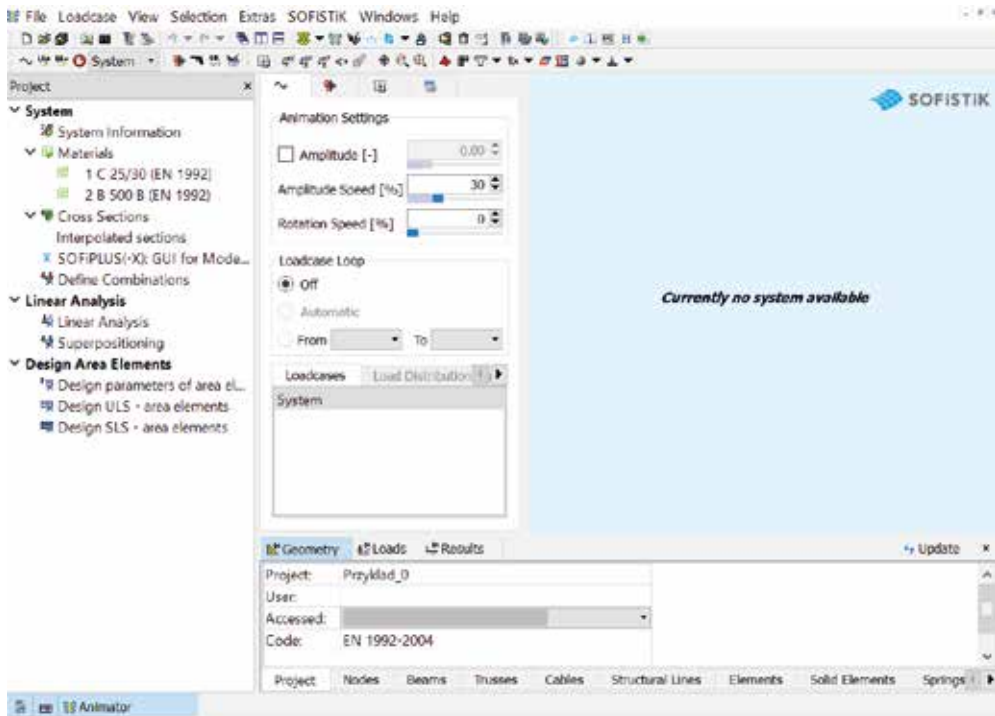

Pierwszą z oferowanych możliwości definicji zadania z perspektywy historii rozwoju środowiska SOFiSTiK i zarazem tą, która pozwala na największą kontrolę wprowadzanych danych, jest moduł *Teddy*  (rys. 2.1). Użytkownik wprowadza tu dane do programu za pomocą kodu tekstowego (*CANDIP*), wywołując poszczególne funkcje za pomocą odpowiednich komend. Mimo że ten sposób wprowadzania danych w wielu programach opartych na MES nie jest już wykorzystywany, programiści SOFiSTiK wciąż doceniają jego możliwości i uniwersalność. Znając język programowania, z łatwością można wprowadzić dane dotyczące geometrii nawet najbardziej skomplikowanych konstrukcji i zlecić wykonanie obliczeń. Moduł ten charakteryzuje się tym, że użytkownik ma możliwość wyboru, czy geometria modelowanej konstrukcji ma zostać wprowadzona fizycznie poprzez definicję węzłów, elementów, warunków brzegowych itd. (*SOFIMSHA*), czy też strukturalnie za pomocą punktów, linii i płaszczyzn charakteryzujących rozwiązywane zadanie (*SOFIMSHC*).



Tablica 2.1


Podstawowa struktura i moduły systemu SOFiSTiK

SOFiSTiK Structural Desktop			
SOFiSTiK Database/ CDB			
PREPROCESSING	PROCESSING	POSTPROCESSING	DATA-EXCHANGE
Wprowadzanie informacji o geometrii do programu	Materiały/ obciążenia/ analiza	Wymiarowanie/ rezultaty/ wizualizacje	Import/ eksport danych
• TEDDY Dane parametryczne	• AQUA Materiały i przekroje	• Design Wymiarowanie	• SOFiCAD Rysunki konstrukcyjne
– TEMPLATE Interaktywne przykłady	• SOFiLOAD Dodawanie obciążeń	– AQB Układy prętowe	• REVIT Structure Koordynacja architekt–konstruktor
• SOFiPLUS Dane graficzne	• TENDON Sprężanie konstrukcji	– AQBS Pełzanie, skurcz, relaksacja	• IFC Otwarty format wymiany danych
• SOFiMSHA Modelowanie strukturalne	• CSM/ CSG Fazy budowy	– BEMESS Płyty i powłoki	• DWG/ DXF Formaty projektowania CAD
• SOFiMSHC Modelowanie geometryczne	• ASE Jądro liczące programu	– BDK Stateczność, zwichrzenie	• ASCII Otwarty format wymiany danych
• WinTUBE Geomechanika, tunele	• DYNA Analizy dynamiczne	• Plotting/ Visualization Rezultaty/ wizualizacje	• VBA, C++, Fortran Języki programowania
	• TALPA Moduł dotyczący geotechniki	– ANIMATOR Narzędzie do wizualizacji	• SteelCON Projektowanie łączników stalowych
	• STAR Układy belkowe 3D	– WINGRAF/ RESULT VIEWER Wyniki graficzne/ tekstowe	
	• HASE Osiedzenia, naprężenia gruntu	– REPORT BROWSER Raporty z obliczeń, błędy	
	• HYDRA Pola magnetyczne, wody gruntowe	– WinG Drukowanie parametryczne	
	• ELLA Obciążenia ruchem ulicznym		
	• SIR Obliczenia sił przekrojowych		
	• MAXIMA Superpozycja		
	• COLUMN/ FOOTiNG Analiza słupów (teoria II rzędu)		
	• DOLFYN Mechanika płynów (CFD)		

Rys. 2.1. Moduł *Teddy* 🐻Rys. 2.2. Moduł *Sofplus* X

Rys. 2.3. Moduł *Structural Desktop* 

Drugim z modułów, znacznie różniącym się od *Teddy* , jest *Sofiplus*  (niezależny program) czy *Sofiplus* (nakładka na AutoCad). Umożliwia on pracę nad stworzeniem modelu konstrukcji w formie graficznej (rys. 2.2). Na podstawie wprowadzonych punktów, linii i płaszczyzn po definicji warunków brzegowych rozwiązywanego zadania program generuje siatkę MES, korzystając z posiadanych algorytmów.

W aktualnej wersji środowiska systemu SOFiSTiK najbardziej uniwersalna jest platforma *Structural Desktop*  (*SSD*) (rys. 2.3), w której użytkownik ma dostęp do wprowadzania danych zarówno za pomocą języka programowania CANDIP, jak i w formie graficznej CAD. Platforma daje możliwość pełnej interakcji pomiędzy wszystkimi modułami środowiska – tymi, które są wykorzystywane jako PREPROCESSOR, PROCESSOR, jak również POST-PROCESSOR.

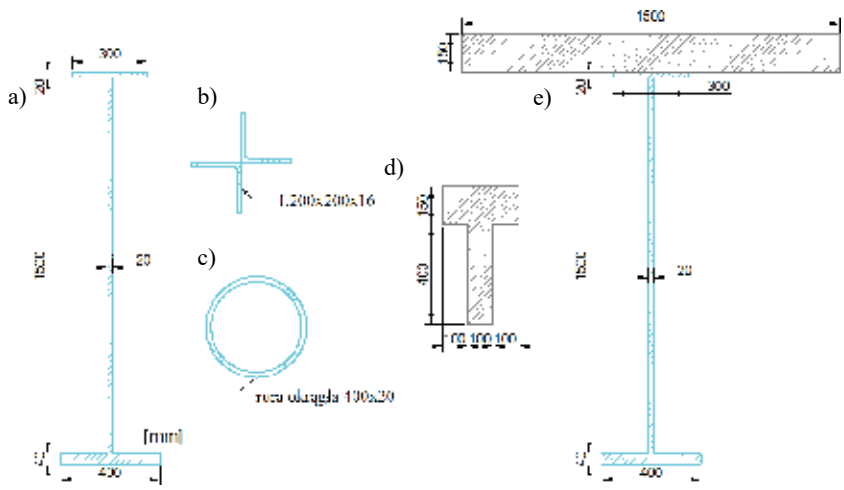
3. UKŁADY PRĘTOWE

W niniejszym rozdziale na prostych przykładach układów prętowych przedstawia się sposób definiowania i prowadzenia fragmentarycznych i końcowych obliczeń w programie SOFiSTiK. Do organizacji wszystkich segmentów tworzących tzw. model obliczeniowy, czyli: materiałów, przekrojów, geometrii oraz obciążeń, zostanie wykorzystany moduł *Teddy* (tablica 2.1).

3.1. Materiały i przekroje

Przykład 3.1. Wykorzystując program *AQUA* (tabl. 2.1), zdefiniować dwa rodzaje materiałów: stal S 355 oraz beton C 35/45. Ponadto stworzyć strukturę danych opisujących sześć przekrojów poprzecznych prętów/ belek (rys. 3.1):

- a) przekrój stalowy typu blachownica;
- b) przekrój stalowy złożony z dwóch profili walcowanych L200×200×16;
- c) przekrój stalowy rurowy;
- d) przekrój betonowy w kształcie litery T;
- e) przekrój zespolony stalowo-betonowy typu T;
- f) przekrój betonowy, którego charakterystyki przekrojowe dane są wprost:
 $A=0,13\text{ m}^2$, $I_y=7,508\times 10^{-3}\text{ m}^4$, $I_z=0,7083\times 10^{-3}\text{ m}^4$.



Rys. 3.1. Przekroje do wprowadzenia w programie

Zdefiniowanie zadania w systemie SOFiSTiK ma formę kodu (w tym skrypcie będzie on wyróżniony czcionką Courier New) zapisanego w tzw. języku problemowo zorien-

towanym rozpoznawanym przez program. Kod ten, interpretowany przez program, należy wpisać bezpośrednio jak tekst do modułu *Teddy* 🗄️ (rys. 2.1). Kod można wzbogacić własnym komentarzem umieszczonym bezpośrednio w czytanej przez program tekście, musi być on jednak poprzedzony symbolem §. Komentarze poprzedzane znakiem § są pomijane przez interpretator programu. Dodatkowo, po prawej stronie kodu poszczególnych komend umieszczono wyjaśnienia umożliwiające ich lepsze zrozumienie. Tych wyjaśnień nie należy wprowadzać do programu.

`+prog aqua`

`head przekroje+materialy`

`echo opt full`

Wywołanie programu *AQUA* (tabl. 2.1) służącego do definiowania materiałów i przekrojów.

Nagłówek (tytuł). Zapisany jednym ciągiem (maks. 72 znaki), nie może zawierać polskich liter i spacji.

Komendą *echo* określa się zakres wyników, które zostaną zapamiętane w bazie systemu i które będą dostępne poprzez *Report Browser* 📄 (tabl. 2.1).

ECHO			
Item	Description	Unit	Default
OPT	A literal from the following list: MAT Material parameters SNO Saving options at section SECT Cross section elements REFP Generated elements and references SDEF Cross section values restart SYST System statistic PICT Properties of Pictures IEQ Edges of the topological analysis WIND Wind coefficients SPRI Spring characteristic curves BORE Bore profile values FULL Select all options	LIT	FULL
VAL	The extent of the output OFF nothing computed / output NO no output YES regular output FULL extensive output EXTR extreme output	LIT	FULL
VAL2	Additional value	-	-

Wartość domyślna

Odwołanie do tekstu poniżej tabeli

Możemy ograniczyć liczbę wyświetlanych informacji. Należy wtedy osobno zdefiniować interesujące nas parametry funkcją *OPT*.

Wartość *FULL* wyświetli wszystkie parametry zdefiniowanych materiałów i przekrojów.

Przykładowe możliwości opcji VAL dla materiałów:
 NO – wyświetli tylko nazwę normy oraz materiału,
 YES – wyświetla dodatkowo parametry materiału.

In case of no ECHO input all options are set to YES (ECHO MAT has NO). The input of the option alone is therefore sufficient for increasing the value to FULL. The record name ECHO must be entered for every record.

MAT

NO Only name of design code and materials
 YES Material constants

Rys. 3.2. Pomoc do programu *AQUA* – funkcje komendy *echo*

W programie dostępna jest interaktywna pomoc/ instrukcja. Aby z niej skorzystać, należy ustawić kursor w wierszu danej komendy, np. *echo*, a następnie uruchomić ją klawiszem *F1*. Po lewej stronie ekranu otworzy się instrukcja w formacie *pdf* (rys. 3.2). Dokument ten od razu zostanie przewinięty do odpowiedniej strony powiązanej z funkcją danej komendy. Aby przenieść się na stronę innej komendy, wystarczy bez zamykania pomocy umieścić kursor w jej wierszu i przewijając strony, odszukać potrzebną informację.

\$ norma

```
norm dc en ndc 1993-2005 coun 00 unit 5
```

Komentarz własny ułatwiający późniejszą orientację w kodzie. Poprzedzony znakiem \$ tekst, aż do zmiany linii, nie będzie interpretowany (zostanie pominięty) przez program.

Wybór normy technicznej, która będzie obowiązująca. Zapis ten wskazuje na eurokod EN 1993–2005. Wybór normy nie wpływa na wartości sił wewnętrznych, jest jednak konieczny, aby m.in. wygenerować dane materiałowe czy np. określone przez normę obwiednie sił wewnętrznych. Dodatkowo, poprzez funkcję *coun* o parametrze *00*, ustala się, że wybrany eurokod jest w wersji ogólnej, tzw. europejskiej. Natomiast przez funkcję *unit* o parametrze *5* doprecyzowano, iż w programie będą stosowane jednostki wg tzw. standardu *Structural Engineering*, czyli wielkości lokalne, np. przekroje wprowadza się w [mm], a wymiary globalne układu w [m].

\$ materialy

```
steer no 1 type s clas 355 $ stal
```

Komentarz własny, poprzedzony znakiem \$.

Definicja materiału o przepisany przez użytkownika numerze, tu nr 1 dla stali S 355. W celu określenia parametrów materiału korzystamy z opcji pomocy (rys. 3.3).

UWAGA. W zapisie poszczególnych komend można pomijać podanie funkcji, pod warunkiem że ich parametry/ wartości wpisuje się w kolejności narzuconej przez pomoc/ instrukcję. Na przykład zapis: *steer no 1 type s clas 355* można skrócić do postaci: *steer 1 s 355*. Jeżeli jednak zachodzi potrzeba wykorzystania funkcji bez zachowania wskazanej w instrukcji kolejności, należy jej określenie pozostawić jako identyfikator. Przykładowo, gdyby zaszła potrzeba zmiany modułu sprężystości (Younga) definiowanego materiału, zapis powinien przyjąć następującą formę: *steer 1 s 355 es 190000*. W początkowym okresie pracy i rozpoznawania programu zaleca się stosowanie pełnych identyfikatorów poszczególnych funkcji.

STEE			
Item	Description	Unit	Default
NO	Material number (1-999)	–	1
TYPE	Type of the material (see remarks)	LIT	*
	S Structural steel		
	B Reinforcing steel		
	Y Prestressing steel		
AL	Aluminium alloy		
CLAS	Steel class or quality	*	*
FY	Yield strength ($f_{0.01}$ or $f_{0.02}$)	N/mm ²	*
FT	Tensile strength	N/mm ²	*
FP	Elastic limit	N/mm ²	*
ES	Elastic modulus	N/mm ²	*
MUE	Poisson's ratio	–	0.3
GAM	Unit weight	kN/m ³	*
ALFA	Thermal expansion coefficient	1/K	*
SCM	Typical material safety factor		*

Nadanie własnego numeru definiowanemu materiałowi

Brak wartości domyślnej

Przypisanie rodzaju stali wg przyjętej normy

Możliwość modyfikacji parametrów przyjętej stali, np.:
 FY – granica plastyczności,
 ES – moduł Younga,
 GAM – ciężar.
 Należy pamiętać o jednostkach NARZUCONYCH przez program, prezentowanych w kolumnie Unit.

Rys. 3.3. Pomoc/ instrukcja do programu AQUA – wybrane funkcje komendy *stee*

```
conc 2 c 35 titl c35/45 $ beton
```

Definicja materiału nr 2, tutaj beton C 35/45. Zasady postępowania są takie same jak w przypadku wcześniej już zdefiniowanej stali. Tutaj dodatkowo nadano materiałowi tytuł c35/45 (musi być to ciąg znaków bez spacji i polskich liter). Zaleca się takie postępowanie, ponieważ tytuł identyfikujący materiał ułatwia orientację w późniejszej pracy z programem.

```
$ przekroje poprzeczne
```

```
sect no 1 mno 1 titl przekroj_a
```

Definicja przekroju nr 1, tutaj blachownicy stalowej (rys. 3.1). Tego typu „własny” przekrój oczywiście nie znajdzie się w bazie programu. Dlatego niezbędne jest wykorzystanie komendy *sect* pozwalającej na wprowadzenie pożądanego kształtu. Dodatkowe funkcje komendy *sect* przedstawiono na rys. 3.4.

```
poly type opz
```

Komenda *poly* umożliwia poligonowy sposób wprowadzenia punktów charakterystycznych kształtu w układzie współrzędnych lokalnych *yz*. Wybierając *type opz*, przekrój poprzeczny tworzy się z wykorzystaniem symetrii względem osi *z*.

SECT			
Item	Description	Unit	Default
NO	Cross section number	-	1
MNO	Material number for cross section	-	1
MRF	Material number of reinforcement	-	*
MRFL	Material link reinforcement	-	*
ALPH	Angle of rotation	Degrees	0
YM	Offset of all cross section ordinates	[mm] ₁₀₁₁	0
ZM	Offset of all cross section ordinates	[mm] ₁₀₁₁	0
FSYM	Suppress rotation of principal axes	LIT	NO
	YES IYZ always set to zero NONE IYZ set to zero when smaller than 0.001·(IY+IZ)		
BTYP	preferred beam type	LIT	BEAM
	CENT centric beam		
	BEAM excentr. with reference axis		
	TRUS only truss (no bending)		
	CABL only cables		
	COMP centric compressive member COLU excentr. column		
BCY	Buckling strain curve for y-y axis	LIT	*
BCZ	Buckling strain curve for z-z axis	LIT	BCY
KTZ	Small part addition (presently not used)	-	*
TITL	Cross section designation	Lit32	-

Tworzonemu przekrojowi należy przyporządkować wcześniej zdefiniowany materiał.

Możliwość obrotu profilu o żądany kąt (w stopniach)

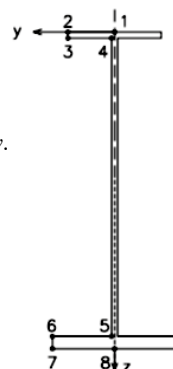
Typowo oś elementu skończonego jest przypisana środkowi ciężkości przekroju. W przypadku konieczności wprowadzenia mimośrodu należy wykorzystać funkcje YM i ZM. Wartości mimośrodu należy podać względem przyjętego początku lokalnego układu współrzędnych przekroju.

Rys. 3.4. Pomoc/ instrukcja do programu AQUA – wybrane funkcje komendy *sect*

```
vert 1 0 0
      2 150 0
      3 150 20
      4 10 20
      5 10 1520
      6 200 1520
      7 200 1560
      8 0 1560
```

Definicja punktów przekroju blachownicy w lokalnym układzie współrzędnych y-z, z wykorzystaniem symetrii względem osi y.

Jeśli tę samą komendę zapisuje się jedną pod drugą jej nazwę można pominąć



UWAGA. Powyższe współrzędne podano w [mm], co wynika z wcześniejszego wyboru parametru *unit* 5 przy definicji normy technicznej. W razie potrzeby wprowadzenia części dziesiętnych należy pamiętać, że oddziela się je kropką, a nie przecinkiem jak w zapisie polskim.

`sect 2 mno 1 titl przekroj_b` Przekrój nr 2, dwa kątowniki (rys. 3.1).

`prof 1 type L 200 16`

Komenda *prof* wykazuje na pobranie informacji o profilach walcowanych z bazy programu (listę ujętych w bazie profili zawiera pomoc/ instrukcja w tabeli poniżej opisu komendy *prof*).


```
prof 2 type L 200 16 alph 180
```

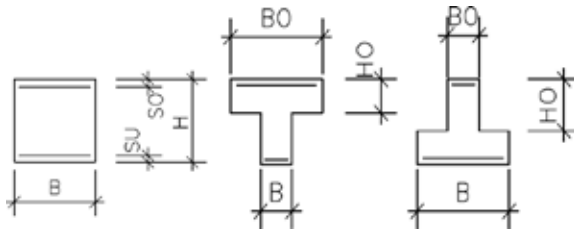
Funkcja *alph* obraca przekrój względem lokalnej osi x o wartość kąta 180 stopni.

```
scit 3 d 400 t 20 mno 1 titl przekroj_c
```

Przekrój nr 3, rura stalowa (rys. 3.1). Komenda *scit* jest samodzielna (nie wymaga komendy *sect*).

```
srec 4 mno 2 h 550 b 100 ho 150 bo 300 titl przekroj_d
```

Przekrój nr 4, beton typu T (rys. 3.1). W instrukcji można znaleźć wyjaśnienie dla poszczególnych parametrów przekroju (rys. 3.5).



Rys. 3.5. Pomoc/ instrukcja do programu *AQUA* – parametry komendy *srec*

```
sect no 5 titl przekroj_e
```

Przekrój nr 5, układ zespolony stalowo-betonowy (rys. 3.1). W przypadku, gdy w jednym przekroju występują dwa materiały, funkcji *mno* nie przypisuje się komendzie *sect*, a przenosimy ją do komendy *poly*. Tutaj komendy *poly* używa się dwa razy, tj. dla części betonowej i stalowej osobno.

```
$ plyta betonowa
```

```
poly type opz mno 2
```

```
vert 1 0 0
      2 750 0
      3 750 -150
      4 0 -150
```

Definicja płyty betonowej o wymiarach 1500×1500 mm z wykorzystaniem symetrii wz. lokalnej osi z .

```
$ blachownica
```

```
poly type opz mno 1
```

```
vert 1 0 0
      2 150 0
      3 150 20
      4 10 20
      5 10 1520
      6 200 1520
      7 200 1560
      8 0 1560
```

Definicja blachownicy z wykorzystaniem symetrii względem lokalnej osi z , lokalizacja punktów wg rysunku jak w przypadku przekroju nr 1.










```
sval 6 mno 2 a 0.13 iy 7.508e-3 iz 0.7083e-3 titl przekroj_f
```

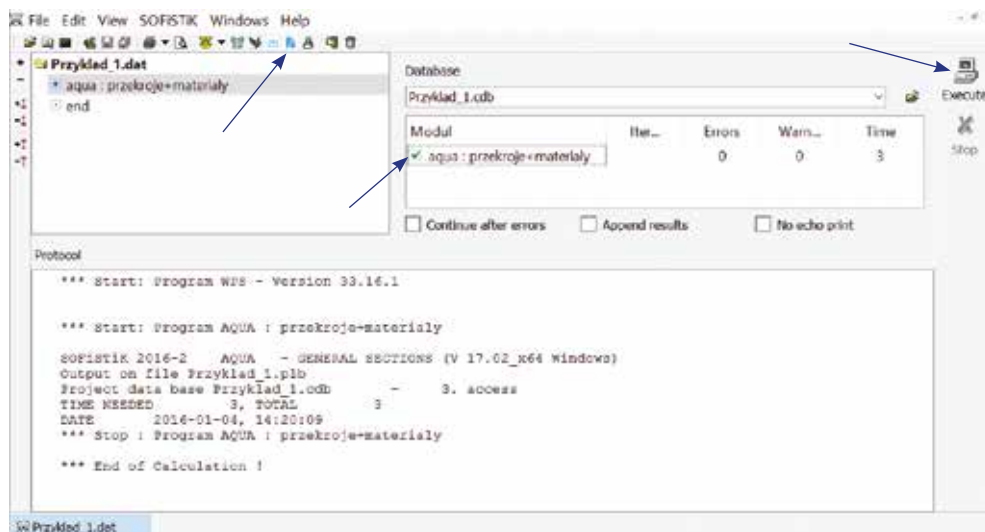
Przekrój nr 6, charakterystyki wprowadzane bezpośrednio w układzie lokalnym elementu: $A=0,13\text{ m}^2$, $I_y=7,508\times 10^{-3}\text{ m}^4$, $I_z=0,7083\times 10^{-3}\text{ m}^4$. UWAGA. W komendzie *sval* parametry przekroju podaje się odpowiednio w m^2 oraz m^4 , patrz

pomoc/ instrukcja do programu *AQUA*, kolumna *Unit*.











end

Każdy program (tutaj *AQUA*) powinien być zakończony komendą *end*.

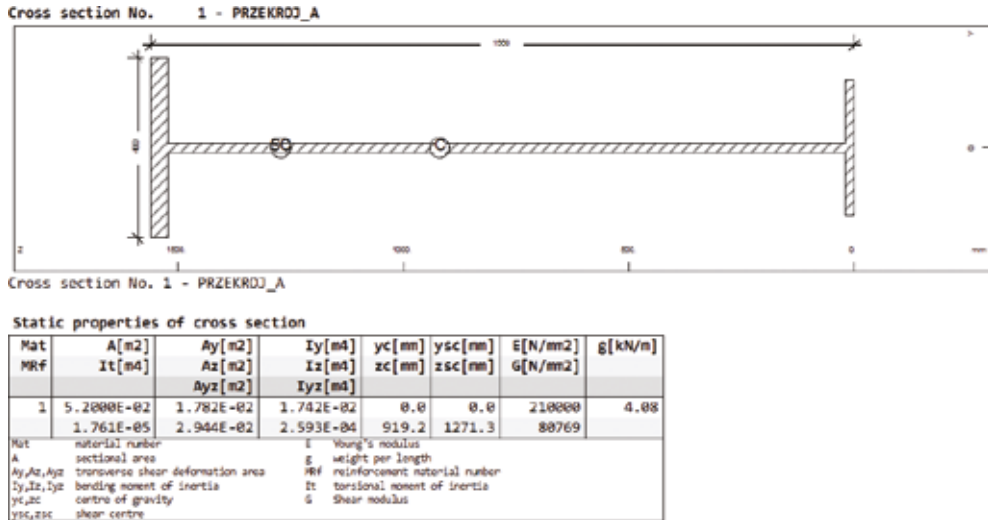
Wprowadzenie do międzymodułowej bazy danych (*CD-BASE*) informacji o materiałach i przekrojach opisanych powyższym kodem oraz sprawdzenie ich poprawności (sygnalizowane ostrzeżeniami  i błędami ) odbywa się poprzez wykonanie obliczeń za pomocą modułu *WPS* . Moduł *WPS*  (rys. 3.6) jest dostępny z poziomu modułu *Teddy*  (rys. 2.1) poprzez kliknięcie ikony  w lewym pasku zadaniowym lub użycie klawisza funkcyjnego *F12*. Obliczenia uruchamia się, klikając w ikonę *Execute* w prawym górnym narożniku okna. Zadanie wykonane poprawnie są sygnalizowane przez symbol , w przypadku ostrzeżenia pojawia się symbol , natomiast błędu – symbol .



Rys. 3.6. Moduł *WPS* – obliczenia programu *AQUA*

W każdej z powyższych sytuacji (poprawności , ostrzeżenia  lub błędu ) możliwa jest kontrola raportu z obliczeń poprzez użycie modułu *Report Browser* . Moduł ten można otworzyć bezpośrednio z poziomu *WPS* , klikając odpowiednią ikonę w górnym pasku okna. Kontrola taka jest zalecana, ponieważ jest to najważniejsza sytuacja do sprawdzenia poprawności wygenerowanych przekrojów (rys. 3.7), ewentualnie ustalenia przyczyny wystąpienia ostrzeżeń  lub błędów . W przypadku ostrzeżeń lub błędów zaleca się dokładne przeanalizowanie komunikatów wygenerowanych przez program w module *Report Browser*  (aby były one łatwo zauważalne, są oznaczone – odpowiednio – zieloną lub czerwoną czcionką). Po ustaleniu przyczyny należy wrócić do modułu *Teddy* , wprowadzić korekty, a następnie jeszcze raz uruchomić moduł *WPS*  i wykonać powtórne obliczenia.

UWAGA. Wystąpienie błędu ✖ uniemożliwia dalszą pracę w programie, wymaga korekty kodu. Wystąpienie ostrzeżenia ⚠ można zignorować, jeśli jego przyczyna nie wpływa na rozwiązanie, jeśli jednak jest znacząca, to wymaga korekty kodu.



Rys. 3.7. Moduł *Report Browser* – fragment raportu dotyczącego przekroju nr 1, tj. a) z rys. 3.1

UWAGA. W obliczeniach charakterystyk przekrojowych skręcania, szczególnie złożonych profili cienkościennych (zamkniętych), ze względu na stosowanie przybliżonej metody w algorytmach modułu *AQUA* mogą się pojawić rozbieżności nawet typu jakościowego w stosunku do wyników teorii skręcania. Z tego względu należy zawsze kontrolować wyliczone przez program położenie środka skręcania, zwłaszcza dla profili symetrycznych i gwiazdzistych, gdzie jego lokalizacja jest znana z założenia.

3.2. Definiowanie geometrii – SOFIMSHA

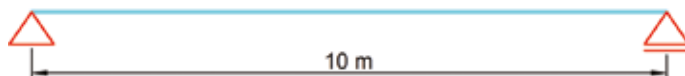
Przed stworzeniem tzw. modelu obliczeniowego w programach wykorzystujących metodę elementów skończonych (i nie tylko) zawsze należy podjąć co najmniej dwie decyzje. Dotyczą one wyboru oferowanego przez program MES typu elementu oraz ustalenia technik tworzenia siatki dyskretyzacji i jej gęstości w kontekście zapewnienia oczekiwanej dokładności rozwiązania. Oczywiście, poza formalną stroną osiągnięcia założonego poziomu dokładności metody numerycznej, poziom oczekiwań od strony technicznej będzie inny w zależności od etapu rozwiązywania problemu inżynierskiego, np. etapu projektu wstępnego, projektu szczegółów konstrukcyjnych czy ekspertyzy badawczej. Decyzja ta należy do użytkownika programu i w dużym stopniu jest uzależniona od jego wiedzy teoretycznej i doświadczenia zawodowego.

W przykładach zawartych w niniejszym rozdziale zostanie wykorzystany tzw. element prętowy/ belkowy 3D o sześciu stopniach swobody w każdym węźle (trzy translacje: $u-X$, $u-Y$, $u-Z$, odpowiadające przesunięciom wzdłuż odpowiednich osi, i trzy parametry rotacyjne: $phi-X$, $phi-Y$, $phi-Z$, odpowiadające małym obrotom względem odpowiednich osi).

Do stworzenia geometrii belek wykorzystany będzie program *SOFIMSHA* (tabl. 2.1). Moduł ten oferuje podstawowy, a zatem najbardziej ogólny wariant wprowadzania informacji o układzie (zwany modelowaniem strukturalnym). Używając tego modułu, użytkownik samodzielnie wprowadza wszystkie dane opisujące węzły i elementy skończone, program *SOFIMSHA* nie ma bowiem wbudowanych zaawansowanych automatycznych generatorów siatek dyskretyzacyjnych.

W celu wprowadzenia danych strukturalnych do systemu SOFiSTiK należy kontynuować zapis kodu stworzonego wcześniej w module *Teddy* (rozdział 3.1). Dalszy kod zadania dotyczący dyskretyzacji należy – jako tekst do nowego programu *SOFIMSHA* – umieścić po komendzie *end* poprzedniego zadania (programu *AQUA*).

Przykład 3.2. Stworzyć model obliczeniowy stalowej belki swobodnie podpartej o długości $L = 10$ m (rys. 3.8) i przekroju blachownicowym nr 1 zdefiniowanym w rozdziale 3.1 wg rys. 3.1a.



Rys. 3.8. Belka swobodnie podparta – schemat statyczny zadania

Przykład 3.2.1. Wariant traktujący belkę jako pojedynczy element skończony

W tym wariantcie – podkreślmy to stanowczo: tylko na potrzeby ilustracyjne postępowania – zostanie zastosowana najprymitywniejsza z możliwych dyskretyzacji (niezalecana w obliczeniach praktycznych). Model belki będzie tworzyć jeden element skończony nr 101 (numeracja jest dowolna, nie musi się zaczynać od 1 i zachowywać kontynuacji), opisany na dwóch węzłach podporowych nr 1 i 2 (rys. 3.9).



Rys. 3.9. Model belki – dyskretyzacja z zastosowaniem jednego elementu skończonego (niezalecana w obliczeniach praktycznych)

```
+prog sofimsha
head geometria
syst 3d
```

Wywołanie modułu/ programu *SOFIMSHA*.
Nagłówek (tytuł).

Komenda *syst* ustala wymiar przestrzeni zanurzenia dyskretyzowanego układu. Wybór przestrzeni zanurzenia jest powiązany z potrzebą użycia odpowiednich elementów skończonych. Opcja *3d* definiuje układ jako strukturę przestrzenną (trójwymiarową w sensie intuicyjnym). Opcja *3d* wymusza wykorzystanie elementów o maksymalnej dopuszczalnej przez system SOFiSTiK liczbie stopni swobody w węźle. Wybór innych możliwości zanurzenia układu przedstawiono na rys. 3.10.

SYST			
Item	Description	Unit	Default
TYPE	The type of the system may be defined:	LIT	!
	3D spatial structures		
	2D Plane frame or pannel		
	2DSS Plane stress		
	2DSN Plane strain		
	2DAX Axial symmetric system		
	2DSL Plane girder or plate bending		

3D – model przestrzenny
 2D – rama płaska
 2DSS – płaski stan naprężenia
 2DSN – płaski stan odkształcenia
 2DAX – osiowa symetria
 2DSL – ruszt i płyta

Rys. 3.10. Pomoc/ instrukcja do programu *SOFIMSHA* – wybrane funkcje komendy *synt*

```
echo full
$ wezly definicja
node no 1 x 0 y 0 z 0
node no 2 x 10 y 0 z 0
```

Definicja pojedynczego węzła (rys. 3.11).

NODE			
Item	Description	Unit	Default
NO	Node number	-/	!
	MESH/IMSH Generation	LIT	
	FIT/FITL Sequence of nodes		
	SBOX All within SBOX		
X	X-coordinate	[m] ₁₀₀₁	0
Y	Y-coordinate	[m] ₁₀₀₁	0
Z	Z-coordinate	[m] ₁₀₀₁	0
FIX	Node constraints	Lit16	*

Wartość bezwzględnie wymagana do działania komendy

Numer węzła

Współrzędne węzła w układzie globalnym

Blokowanie stopni swobody – przypisanie do węzła podpory

Rys. 3.11. Pomoc/ instrukcja do programu *SOFIMSHA* – wybrane funkcje komendy *node*

```
$ elementy definicja
beam no 101 na 1 ne 2 ncs 1
```

Definicja pojedynczego elementu typu *beam* (rys. 3.12). UWAGA. Numery węzłów podane po funkcjach *na* i *ne* definiują zwrot lokalnej osi *x* elementu. Definicję osi *y* oraz *z* można pozostawić domyślną, ewentualnie jeżeli zaproponowana definicja jest niepoprawna, skorygować funkcją *kr*, węzłem lub wektorem kierunkowym. Należy mieć to na względzie, ponieważ lokalny układ współrzędnych jest wykorzystywany do definicji charakterystyk przekrojowych i sił wewnętrznych pręta.

BEAM			
Item	Description	Unit	Default
NO	Element number or option	-/LIT	
NA	Number of start beam node	-	!
NE	Number of end beam node	-	!
NCS	Cross section number or section_start.section_end or as formula #(formula) e.g. #(100+#1)	-	!
DIV	Equal partitioning of beam	-	-
NM	optional 3rd subdivision node (see comments in record NODE)	-	-
AHIN	Input of hinge at start node	LIT	-
EHIN	Input of hinge at end node	LIT	-
NP	Number of a bedding profile or generation of the reference axis for an eccentric beam	-	-
BETY	Buckling length coeff. (main bending)	-	0
BETZ	Buckling length coeff. (transverse)	-	BETY
ALFY	Span length coeff. (main bending)	-	-
ALFZ	Span length coeff. (transverse)	-	ALFY
KR	Orientation of local coordinates	LIT/-	0
DRX	Direction vector	-	*
DRY		-	*
DRZ		-	*
DROT	Additional rotation about beam axis	degree	0

Rys. 3.12. Pomoc/ instrukcja do programu *SOFIMSHA* – wybrane funkcje komendy *beam*

\$ podpory, warunki brzegowe
 node 1 - - - fix ppmx

2 - - - fix xp
 end

Nałożenie więzów na węzeł (warunków brzegowych, podporowych) można przeprowadzić bezpośrednio przy jego definicji (rys. 3.11), zwykle jednak wygodniej jest zrealizować ten krok później. Aby dodać informację do zdefiniowanego już węzła (komenda *node*), w miejscu zapisu współrzędnych (x,y,z) wstawia się myślniki. Wartości funkcji *fix* (rys. 3.13) podaje się bez spacji. W zadaniu oprócz wszystkich translacji w węźle 1 zablokowano rotację wokół osi X . UWAGA. Cały układ, postrzegany jako bryła sztywna, ma sześć globalnych stopni swobody (model stworzony jako 3D), stąd wszystkie stopnie niebiorące udziału w deformacji należy zablokować. W tym przypadku blokada obrotu względem osi X zapobiegnie sztywnej rotacji belki traktowanej jako całość wokół tej osi.

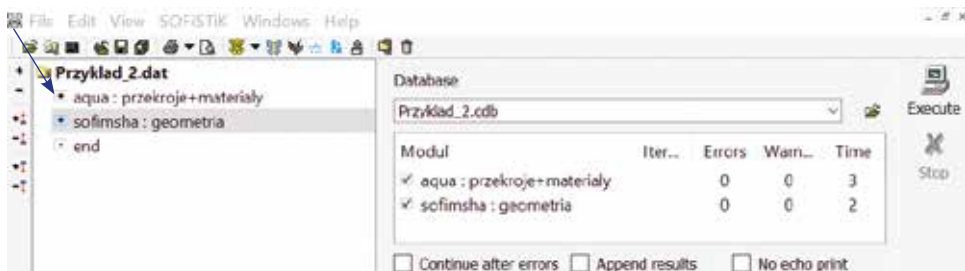
Jeśli tę samą komendę zapisuje się jedna pod drugą, jej nazwę można pominąć.
 Zakończenie programu *SOFIMSHA*.

PX	Constraint of displacement in x	Blokada wybranych translacji
PY	Constraint of displacement in y	
PZ	Constraint of displacement in z	
PR	Skewed constraint of displacement in a direction	
PT	skewed constraint of displacements perpendicular to a direction	
MX	Constraint of rotation about x	Blokada wybranych rotacji
MY	Constraint of rotation about y	
MZ	Constraint of rotation about z	
MR	Skewed constraint of rotation about a direction	
MT	Skewed constraint of rotation perpendicular to a direction	
MB	Constraint of warping	
XP	= PY + PZ	Przypadki mieszane, wprowadzanie więzów na kilku translacjach lub rotacjach
YP	= PX + PZ	
ZP	= PX + PY	
PP	= PX + PY + PZ	
XM	= MY + MZ	Pełne utwierdzenie węzła
YM	= MX + MZ	
ZM	= MX + MY	
MM	= MX + MY + MZ	
FREE	= Deletion of all constraints	
F	= PP + MM + MB	

Rys. 3.13. Pomoc/ instrukcja do programu *SOFIMSHA* – ustawienia funkcji *fix* komendy *node*



Aby sprawdzić, czy geometria została zadeklarowana poprawnie, oraz wprowadzić informację o niej do bazy danych (*CD-BASE*), stworzony kodem ciąg instrukcji należy przeliczyć w module *WPS* (rys. 3.14).

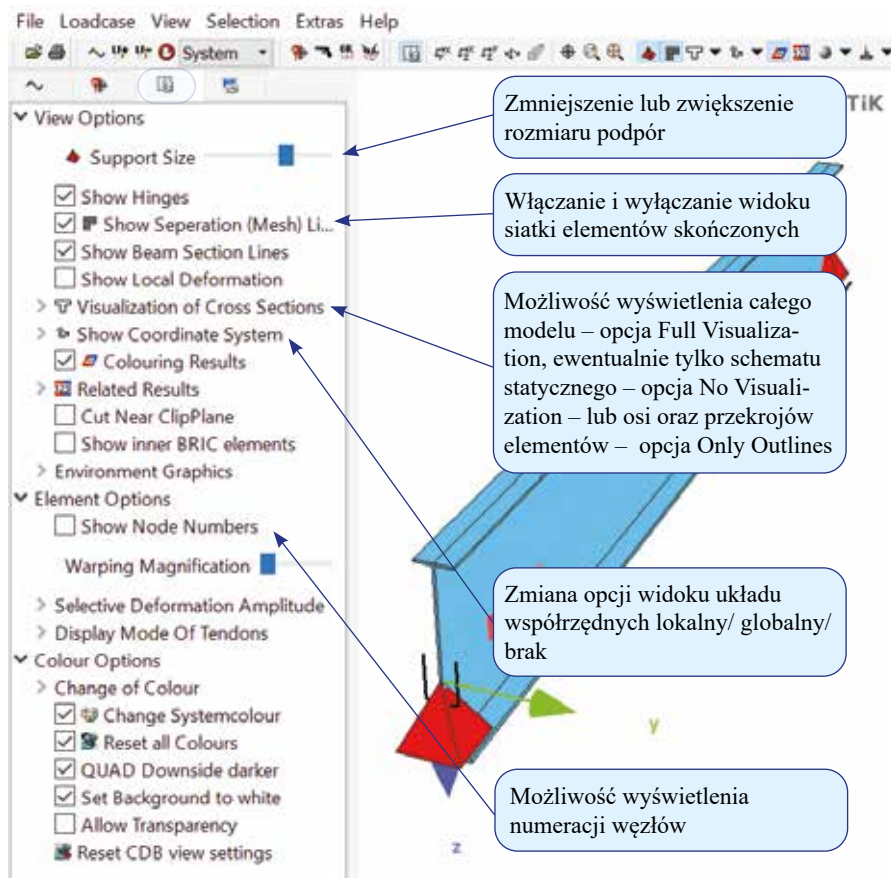
UWAGA. Jeśli w poprzednim kroku został przeliczony (program *AQUA*), nie wprowadzono w nim zmian (a więc informacje w bazie danych *CD-BASE* są aktualne) i nie zmieniono w międzyczasie nazwy pliku modelu, w oknie dialogowym drzewka przykładu (rys. 3.14) można wyłączyć ponowne jego obliczenia (symbol minus), pozostawiając aktywne obliczenia programu *SOFIMSHA* (symbol plus). Takie postępowanie w przypadku rozbudowanych zadań może znacząco skrócić czas oczekiwania na rozwiązanie.




Rys. 3.14. Moduł *WPS*, obliczenia programu *SOFIMSHA*

Jeśli obliczenia zakończyły się poprawnie ✓ (program nie zgłosił ostrzeżeń ✗ lub błędów ✗), wygenerowany model obliczeniowy można przeanalizować wizualnie w module

Animator  (tabl. 2.1). Program ten można otworzyć bezpośrednio w module *WPS* , klikając odpowiednią ikonę w górnym pasku okna. Na rys. 3.15 przedstawiono wizualizację belki z pokazanym globalnym układem współrzędnych związanym z przestrzenią pracy. Ustawienia widoku można zmieniać w trzeciej zakładce okna po lewej. Na rys. 3.15 zaprezentowano wybrane możliwości.



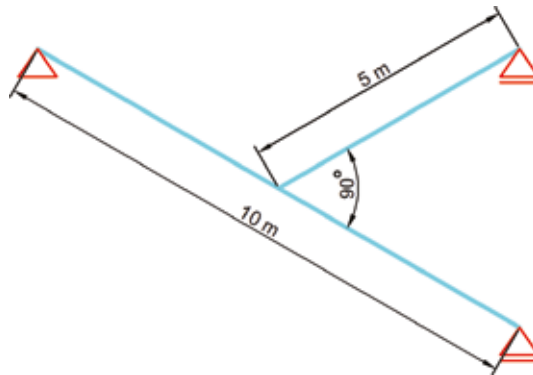
Rys. 3.15. Moduł *Animator* – wizualizacja tzw. modelu obliczeniowego

Należy pamiętać, że moduł *Animator*  jest programem do wizualizacji i obraz w nim prezentowany (np. deformacja konstrukcji) nie zawsze jest wynikiem obliczeń; zdarza się, że jest on wzbogacony tak, aby był przyjazny w odbiorze.

Zadanie 1. Modyfikacja geometrii belki do rusztu

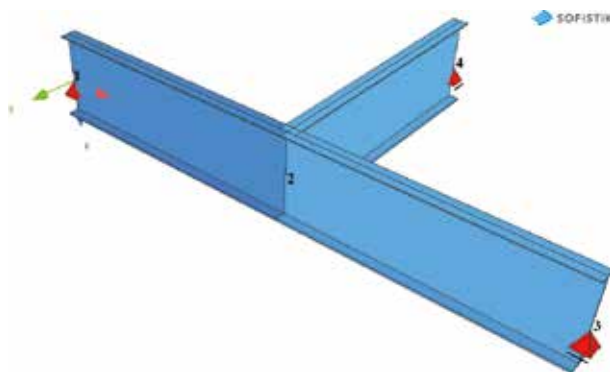
Rozbudować wprowadzoną w poprzednim zadaniu geometrię belki swobodnie podpartą do rusztu blachownicowego zgodnie ze schematem przedstawionym na rys. 3.16. W stosunku do schematu z rys. 3.8 należy w środku belki, w płaszczyźnie XY , dołączyć poprzeczną belkę pod kątem 90° . W tym „treningowym” przykładzie zostanie zachowana najprostsza z moż-

liwych dyskretyzacja po jednym elemencie skończonym w poszczególnych gałęziach rusztu (niezalecana w obliczeniach konstrukcji), z jednoczesnym uwzględnieniem fundamentalnej zasady MES, że elementy skończone muszą się ze sobą łączyć w węzłach dyskretyzacyjnych!



Rys. 3.16. Ruszt – schemat statyczny

Efekt końcowy przedstawiono w module *Animator* na rys. 3.17. UWAGA. W przypadku rusztu swobodnie podpartego należy odblokować wprowadzony wcześniej warunek brzegowy na obrót belki względem osi X .



Rys. 3.17. Ruszt – wizualizacja modelu obliczeniowego rusztu z numeracją siatki węzłów

Przykład 3.2.2. Belka swobodnie podparta – zagęszczanie siatki dyskretyzacyjnej

Wygenerować dyskretyzację belki swobodnie podpartej z rys. 3.8 z podziałem na dziesięć elementów skończonych, rozpiętych na jedenastu równoodległych węzłach (rys. 3.18).



Rys. 3.18. Model belki – zagęszczona siatka dyskretyzacyjna

Do stworzenia geometrii również zostanie użyty program *SOFIMSHA*. Tym razem, aby uniknąć uciążliwego wypisywania współrzędnych wszystkich węzłów, wykorzystamy prosty generator liniowy, wbudowany w moduł *SOFIMSHA*. Algorytm ten tworzy szereg danych na zasadzie proporcjonalnych przyrostów. Opisy odpowiednich szeregów są ujmowane w nawiasy (...), w których umieszcza się pierwszy i ostatni element szeregu oraz wartość przyrostu. Początek kodu opisu zadania jest identyczny jak w przykładzie 3.2.1; zmiany należy wprowadzić w miejscu definicji węzłów i elementów.

```
$ węzły definicja
node no (1 11 1) x (0 1) y 0 z 0
```

Algorytm tworzy węzły o nr. od 1 do 11 z krokiem 1 (tj.: 1, 2, ..., 11).

Położenie węzłów, począwszy od $X = 0$, zmienia się z każdym kolejnym węzłem o krok $\Delta X = 1$ [m] (czyli węzeł 1: $X = 0$ [m], węzeł 2: $X = 0 + 1 = 1$ [m] itd.).

Wszystkie węzły mają jednakowe współrzędne $Y = Z = 0$ (stąd brak nawiasów w zapisie).

```
$ elementy definicja
beam no (101 110 1) na (1 1) ne (2 1) ncs 1
```

Algorytm tworzy elementy od nr. 101 do 110 z krokiem 1 (tj.: 101, 102, ..., 110).



Numer pierwszego węzła pierwszego elementu: 1. Każdy kolejny element zaczyna się w węźle o numerze o 1 większym od poprzedniego (tzn. element 101 zaczyna się w węźle 1, element 102 w węźle $1+1=2$ itd.).

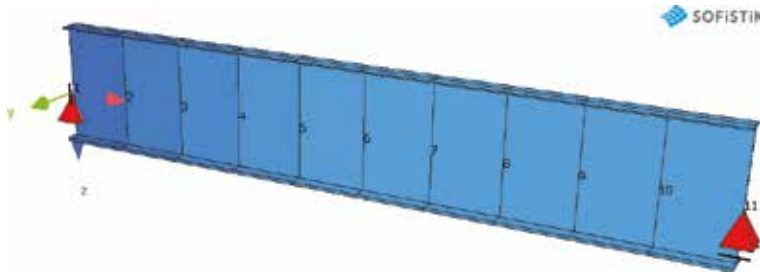
Numer drugiego węzła pierwszego elementu: 2. Każdy kolejny element kończy się w węźle o numerze o 1 większym od poprzedniego (tzn. element 101 kończy się w węźle 2, element 102 w węźle $2+1=3$ itd.).

```
$ podpory
node 1 - - - fix ppmx
     11 - - - fix xp
```

Należy zmienić numer węzła; w tym wypadku ostatnim jest węzeł 11.

```
end
```

Nowy kod geometrii należy sprawdzić i wprowadzić do bazy danych (*CD-BASE*), uruchamiając moduł *WPS* . Odpowiednią wizualizację uzyskujemy poprzez moduł *Animator*  (rys. 3.19).



Rys. 3.19. Belka o zagęszczonym podziale na elementy – wizualizacja

3.3. Obciążenia, obliczenia statyczne

Wykorzystując model obliczeniowy opisany w rozdziale 3.2 jako przykład 3.2.2, wygenerować następujące obciążenia działające na belkę:

- przykład 3.3a, ciężar własny (mnożnik obciążenia 1);
- przykład 3.3b, siła skupiona w $\frac{1}{2} L$ belki o wartości 20 kN oraz w $\frac{3}{4} L$ belki o wartości 10 kN (mnożnik obciążenia 1);
- przykład 3.3c, obciążenie liniowe rozłożone na długości belki w przedziale $[0, \frac{1}{2}] L$ belki o wartości zmieniającej się od 5 do 10 kN/m oraz w przedziale $[\frac{3}{4}, 1] L$ belki o wartości stałej 15 kN/m (mnożnik obciążenia 1);
- przykład 3.3d, obciążenie łączone z powyższych przypadków, przemnożone przez współczynnik 1,1 dla ciężaru własnego (a) i 1,3 dla pozostałych (b i c).

Do definicji oddziaływań statycznych, dynamicznych oraz bardziej szczegółowych, jak np. obciążenie wiatrem w systemie SOFiSTiK, wykorzystuje się program *SOFILOAD* (tabl. 2.1). Należy pamiętać, że program *SOFILOAD* jest tylko generatorem obciążenia. Obliczenia związane z różnym typem analiz (statycznych, dynamicznych, liniowych, nieliniowych) są prowadzone w programie *ASE* (tabl. 2.1). Wyjątkiem od tej reguły jest obciążenie ciężarem własnym, które jako jedyne można wygenerować, a następnie obliczyć w programie *ASE*. Wszystkie obciążenia generowane w programie umieszczają się w przypadkach obciążenia (*Load Case – LC*), którym każdorazowo nadaje się numer.

Przykład 3.3a. Należy kontynuować zapis kodu stworzony w module *Teddy* w rozdziale 3.2 (belka o zagęszczonym podziale, przykład 3.2.2). Poniższy kod należy zapisać po komendzie *end* powiązanej z modułem *SOFIMSHA*. Wprowadzanie obciążeń zaczniemy od uwzględnienia ciężaru własnego w module *ASE* (domyślnie liczona jest statyka liniowa).

```
+prog ase
head cw
$ ciezar wlasny
lc 1 dlz 1 titl obc_A_cw
```

Wywołanie programu *ASE*.

Nagłówek (tytuł).

Za pomocą komendy *lc* tworzy się przypadek obciążenia (rys. 3.20), tutaj o numerze 1. Nadając funkcji *dlz* wartość mnożnika obciążenia 1, określa się przypisanie do rozważanego przypadku 100% udziału ciężaru własnego działającego

w kierunku globalnej osi Z. UWAGA. W SOFiSTi-Ku domyślnie globalna oś Z jest skierowana w dół, stąd – przy założeniu takiego kierunku grawitacji – mnożnik obciążenia ciężarem własnym w tym przypadku ma wartość dodatnią.
Zakończenie programu ASE.

end

Item	Description	Unit	Default
NO	Load case number	–	!
	ALL All load cases		
	TEST Instability test		
FACT	Factor for all loads	–	1.0
DLX	Dead weight factor in global X direction	–	0.0
DLY	Dead weight factor in global Y direction	–	0.0
DLZ	Dead weight factor in global Z direction	–	0.0



LC



Numer przypadku obciążenia



Możliwość wprowadzenia mnożnika dla obciążeń

Aktywowanie ciężaru własnego na kierunku globalnym X, Y lub Z

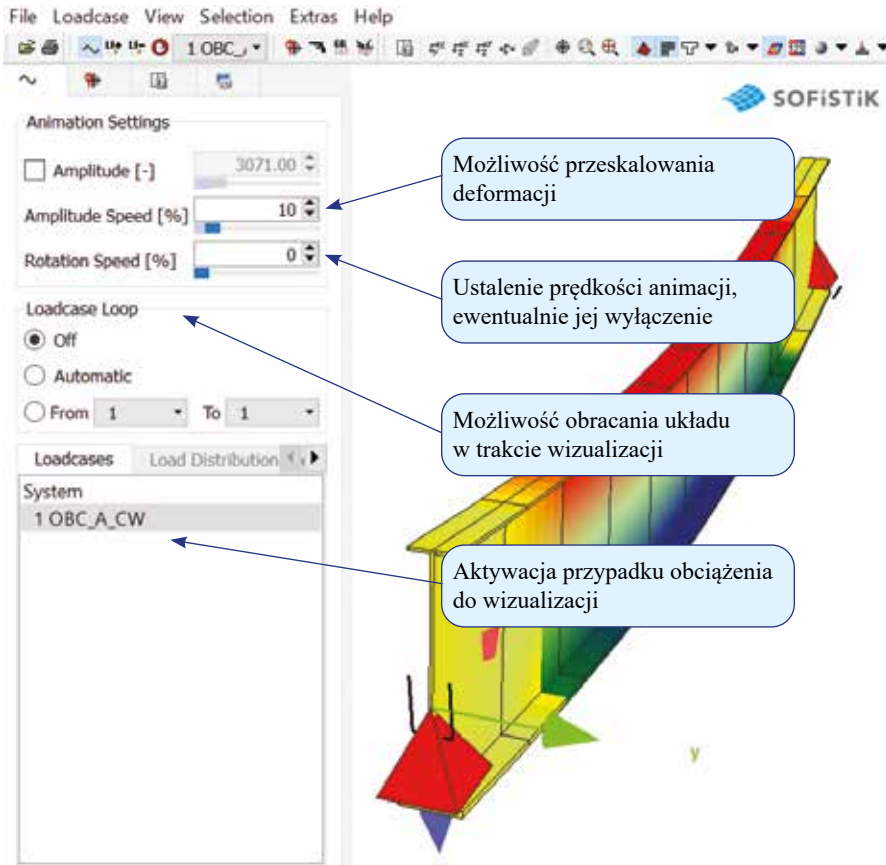
Rys. 3.20. Pomoc/ instrukcja do programu SOFILOAD – wybrane funkcje komendy lc

Aby wykonać obliczenia, należy uruchomić program ASE w module WPS . Animację i wynik deformacji układu uzyskuje się w module Animator  (rys. 3.21).

UWAGA. Jeśli program ASE sygnalizuje błąd , nie musi on wynikać z błędnego zapisu ostatniej części kodu. Przyczyna może wystąpić wcześniej, np. w błędnej geometrii, warunkach brzegowych (brak blokady ruchu sztywnego całego układu lub jego części). W takim przypadku trzeba się dokładnie zapoznać z opisem w Report Browser .

Do zdefiniowania innych obciążeń wykorzystuje się program SOFILOAD. Można to zrobić na dwa sposoby. Pierwsza grupa narzędzi dotyczy tzw. obciążenia typu strukturalnego. Przypisuje się je do konkretnych węzłów lub elementów skończonych już zdyskretyzowanego układu/ struktury. Podejście to jest mniej elastyczne, jednak możliwość popełnienia pomyłki jest tu zdecydowanie mniejsza. Druga grupa narzędzi dotyczy tzw. obciążenia typu wolnego. W tym przypadku obciążenia są definiowane w przestrzeni niezależnie od dyskretyzacji struktury, a program sam poszukuje węzłów lub elementów, do których je przypisze. Dodatkowo możliwe jest tu zdefiniowanie marginesu, w obrębie którego węzły lub elementy zostaną obciążone. Sposób ten jest bardzo wygodny, wymaga jednak precyzji i uwagi, ponieważ możliwość popełnienia błędu jest większa (np. obciążenie niepełne, zdublowane itp.). Ważne jest w tym przypadku dokładne przeanalizowanie ewentualnych ostrzeżeń  w Report Browser .

UWAGA. Powyższych sposobów wprowadzania obciążenia nie należy mylić z pojęciem typu obciążenia występującym w analizie nieliniowej, gdzie obciążenie może zależeć od deformacji układu (obciążenia śledzące) i jego lokalizacji w przestrzeni fizycznej (np. wartość parcia cieczy zależna od zmieniającej się głębokości zanurzenia układu).



Rys. 3.21. Moduł *Animator* – wizualizacja układu obciążonego ciężarem własnym

Przykład 3.3b. Wprowadzenie sił skupionych.

```
+prog sofiload
head obciazenia
lc 2 titl obc_B_skupione
node no 6 type pzz p1 20
```

Wywołanie programu *SOFILOAD*.

Deklaracja obciążenia nr 2, siłami skupionymi. Siła skupiona w środku belki o wartości 20 [kN], traktowana jako obciążenie typu strukturalnego bezpośrednio w węzle nr 6 (rys. 3.22). Symbol *pzz* określa siłę skupioną w kierunku globalnym Z (ważne, aby zwrócić uwagę na jednostki, zależą one od typu). Wybranie innej możliwości zaprezentowano na rys. 3.23.

```
poin ref bgrp type pzz p 10 x 7.5
```

Siła skupiona o wartości 10 [kN] w lokalizacji przestrzennej odpowiadającej długości $\frac{1}{4} L$ belki, traktowana jako obciążenie typu wolnego

(rys. 3.24). Funkcja *ref* służy do ustalenia rodzaju obiektu, jaki ma być „poszukiwany” w celu powiązania z nim obciążenia. Tutaj wybrano *bgrp*, co oznacza, że program przydzieli obciążenie do elementów typu *beam* (w tak prostym przykładzie jak analizowana belka można rozważyć pozostawienie domyślnego *auto*). W zapisie ustalono pozycję obciążenia w punkcie przestrzeni odpowiadającym współrzędnej $X = 7,5$ [m]. Współrzędne Y i Z pominięto, ponieważ ich wartość domyślna wynosi 0 (kolumna *Default* w pomocy).

NODE			
Item	Description	Unit	Default
NO	Number of node or designation of secondary group number	← /LIT	!
TYPE	Type of Loading	LIT	PG
P1	Load values	[*]	!
P2	up to seven depending on load type	[*]	-
P3		[*]	-
P4		[*]	-
P5		[*]	-
P6		[*]	-
P7		[*]	-
FRAD	Radial factor for load values R multiply with radius RR multiply with square of radius	LIT	-
EXX	Exzentricity to node	[m] ₁₀₀₁	-
EYY	(will be immediately converted to moments)	[m] ₁₀₀₁	-
EZZ		[m] ₁₀₀₁	-

Numer węzła

Typ obciążenia

Wartość obciążenia

Dodatkowe wartości (składowe) dla obciążeń definiowanych jako wektory

Możliwość wprowadzenia mimośrod

Rys. 3.22. Pomoc/ instrukcja do programu *SOFILOAD*
– funkcje komendy *node*

TYPE	Description	Unit
PG	Load in gravity direction	kN
PXX	Load in global X-direction	kN
PYY	Load in global Y-direction	kN
PZZ	Load in global Z-direction	kN
MXX	Moment about global X-direction	kNm
MYX	Moment about global Y-direction	kNm
MZZ	Moment about global Z-direction	kNm

Siła skupiona w kierunku ustalonej grawitacji

Siła skupiona w wybranym kierunku układu globalnego

Moment skupiony w wybranym kierunku układu globalnego

Rys. 3.23. Pomoc/ instrukcja do programu *SOFILOAD*
– przykłady typów obciążeń w komendzie *node*

POIN			
Item	Description	Unit	Default
REF	Reference type	LIT	AUTO
	AUTO Automatic		
	MOVE Automatic moving load		
	LAR load distribution area		
	SLN Structural line		
	SAR Structural region		
	GFA Surface number of BRICs		
	NODE generated node		
	BGRP Group of beam elements		
	QGRP Group of QUAD elements		
VGRP Group of BRIC elements			
NO	Reference or group number	-	-
TITL	Title of loading	LIT12	-
PROJ	Projection of load point	LIT	N
	N normal of reference area (GAR)		
	XX projection in global X-direction		
	YY projection in global Y-direction		
	ZZ projection in global Z-direction		
WIDE	Range in direction of projection	[m]1001	0.0
	Default 0.0 with the tolerance = element thickness		
NREF	Reference node for coordinate values	-	-
TYPE	Load type and direction	LIT	PG
	P Load value	*	0.
	X Global coordinates of loading	[m]1001	0.
	Y	[m]1001	0.
	Z	[m]1001	0

Rys. 3.24. Pomoc/ instrukcja do programu *SOFILOAD* – funkcje komendy *poin***Przykład 3.3c.** Wprowadzenie obciążeń rozłożonych.

`lc 3 titl obc_C_rozlozone` Deklaracja obciążenia nr 3, tu obciążenia rozłożonego.

`beam 101 105 1 type pzz pa 5 pe 10`

Obciążenie rozłożone (trapezowe) o wartości zmieniającej się liniowo od 5 do 10 [kN/m], traktowane jako obciążenie typu strukturalnego (rys. 3.25) działające bezpośrednio na elementy skończone o nr. od 101 do 105 co 1, tzn. na odcinku $[0, \frac{1}{2}] L$ belki.

BEAM			
Item	Description	Unit	Default
FROM	Number of first element or GRP	-	!
TO	Number of last element or group number	-	FROM
INC	Increment of element numbers	-	1
TYPE	Load type and direction	LIT	PG
PA	Load value at load start point	[*]	!
PE	Load value at load end point	[*]	PA

Rys. 3.25. Pomoc/ instrukcja do programu SOFILOAD – wybrane funkcje komendy *beam*

LINE,CURV			
Item	Description	Unit	Default
REF	Reference type AUTO Automatic MOVE Automatic moving load LAR load distribution area SLN Structural line SAR Structural region GFA Surface number of BRICs EDG Nodal sequence BGRP Group of beam / cable / truss elements QGRP Group of QUAD elements VGRP Group of BRIC elements	LIT	AUTO
NO	Reference or group number	-	-
TITL	Title of loading	LIT12	-
PROJ	Projection of loading line N normal of reference area (GAR) XX projection in global X-direction YY projection in global Y-direction ZZ projection in global Z-direction XXL as XX, sign for local component YYL as YY, sign for local component ZZL as ZZ, sign for local component	LIT	N
WIDE	Range in direction of projection Default 0.0 with the tolerance = element thickness	[m]1001	0.0
NREF	Reference node for coordinate values	-	-
TYPE	Load type and direction	LIT	PG
P1	Load value of first point	*	0.
X1	Coordinates of first point or input of SPT	[m]1001	0.
Y1	for X1 and the point number for Y1 or	[m]1001	0.
Z1	S or SX and the ordinate along a SLN.	[m]1001	0.
P2	Load value of second point	*	*
X2	Coordinates of second point or input of SPT	[m]1001	X1
Y2	for X2 and the point number for Y2 or	[m]1001	Y1
Z2	S or SX and the ordinate Y2 along a SLN.	[m]1001	Z1
...			

Rys. 3.26. Pomoc/ instrukcja do programu SOFILOAD – funkcje komendy *line/curv*


```
line ref bgrp type pzz p1 15 x1 7.5 x2 10
```

Obciążenie rozłożone (stałe) o wartości 15 kN/m, traktowane jako obciążenie w kierunku globalnym Z typu wolnego (rys. 3.26) działające na osi X w przedziale [7,5; 10] m. Wartość obciążenia zdefiniowano funkcją *p1*, natomiast pozycję w przestrzeni – funkcjami *x1* i *x2*. Współrzędne *y1*, *z1* i *y2*, *z2* pominięto, ponieważ ich wartość domyślna wynosi 0 (kolumna *Default* w pomocy).

```
end
```


Zakończenie programu *SOFILOAD*.

Powyższe działania kończą etap wprowadzania danych zadania do systemu i ich częściowej weryfikacji. Kolejnym krokiem jest faza obliczeń, którą przeprowadza się w module ASE.

```
+prog ase
head obliczenia
lc 2,3
```

Aby wykonać obliczenia dla przypadków 2 oraz 3, w programie *ASE* wystarczy wpisać ich numery po komendzie *lc*. Numery na liście kolejnych przypadków oddzielamy przecinkiem. Można również zastosować zapis z nawiasem (*2 3 1*), który wskazuje na użycie generatora liniowego i jest interpretowany przez program następująco: od 2 do 3 co 1.

```
end
```

Obliczenia programu *SOFILOAD* i *ASE* uruchamia się przy wykorzystaniu modułu *WPS*  (rys. 3.14).

Przykład 3.3d. Przypadek obciążenia nr 4 zostanie utworzony jako kombinacja liniowa/ superpozycja obciążeń nr 1, 2 i 3 z odpowiednimi mnożnikami.

```
+prog ase
head kombinacja
lc 4 dlz 1.1 titl obc_D_suma
```

```
lcc 2,3 fact 1.3
```

Utworzenie nowego przypadku obciążenia nr 4. Ciężar własny (*dlz*, przykład 3.3a) musi zostać zadany jeszcze raz (nie podlega bowiem kopiowaniu), tym razem powiększony o 10%, mnożnik 1.1. Za pomocą komendy *lcc* do utworzonego ostatnią komendą przypadku (*lc*) można dodać stworzone wcześniej warianty obciążeń z odpowiednim mnożnikiem określonym funkcją *fact*.

```
end
```

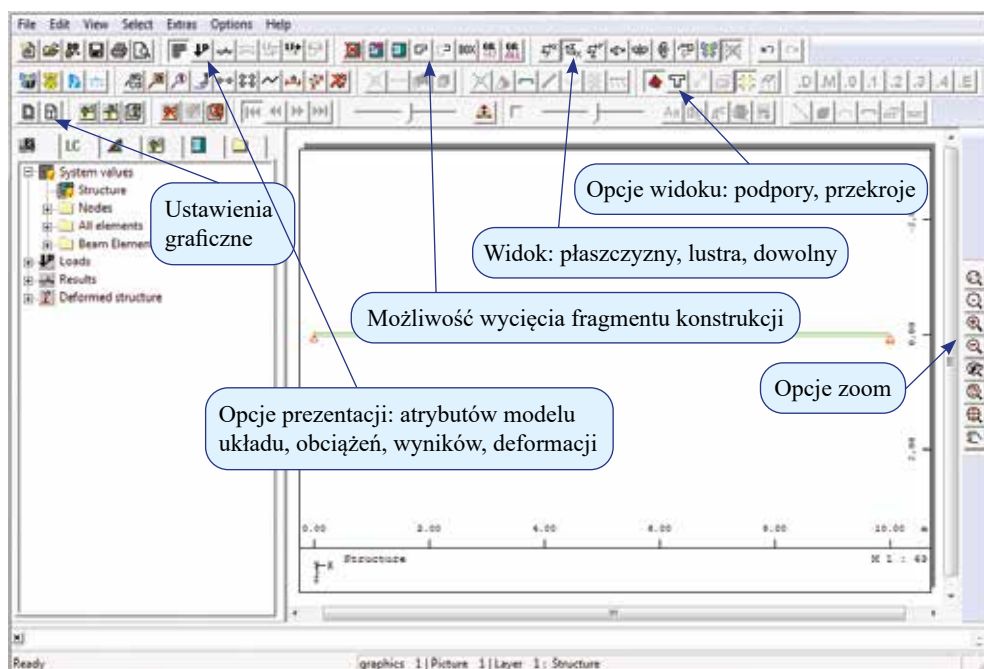
Po wykonaniu obliczeń przechodzi się do najtrudniejszej – z punktu widzenia konstruktora – fazy, mianowicie interpretacji i oceny poprawności uzyskanych wyników. Faza ta wymaga doświadczenia i wiedzy teoretycznej z zakresu wytrzymałości materiałów, mechaniki budowli, dynamiki budowli, teorii sprężystości i plastyczności, stateczności konstrukcji, mechaniki ośrodków ciągłych itd. oraz oczywiście samej metody elementów skończonych.

3.4. Prezentacja wyników – atrybuty modelu obliczeniowego, przemieszczenia, siły

Program SOFiSTiK stawia do dyspozycji cztery moduły, za pomocą których można wizualizować wyniki (raporty, szczegóły modelowania układu oraz rezultaty obliczeń). Dwa z nich zostały już wcześniej użyte w zadaniach cząstkowych, są to *Report Browser* oraz *Animator*. Kolejnymi są *WinGraf* oraz *Result Viewer*. Moduł *WinGraf* jest przeznaczony do prezentacji wyników w postaci graficznej, np. listowanie wartości na bazie schematu statycznego, wykresy, mapy itp. Za pomocą programu *Result Viewer* w wygodny sposób można zestawiać wyniki w postaci tabelarycznej. Podstawowe zasady działania modułów zostaną zademonstrowane na przykładzie z Rozdziału 3.3. Dostęp do wszystkich czterech modułów jest możliwy z poziomu okna *Teddy* lub modułu *WPS* (ikony w lewym bocznym pasku) lub modułu *WPS* (ikony w pasku górnym).

Graficzna prezentacja wyników

Do graficznej prezentacji wyników będzie użyty moduł *WinGraf*. Ekran programu oraz jego podstawowe funkcje przedstawiono na rys. 3.27. Konfiguracja pasków narzędziowych jest elastyczna, można je dowolnie dostosować do własnej wizji, przeciągając je w docelowe miejsce. W przypadku braku któregoś z potrzebnych pasków w domyślnej wersji okna wystarczy kliknąć prawym klawiszem myszy w odpowiednią ikonę i go włączyć.



Rys. 3.27. Ekran modułu *WinGraf*

Największy obszar ekranu zajmuje okno graficzne, w którym po otwarciu modułu *WinGraf* widoczny jest schemat statyczny analizowanego układu. Okno graficzne otaczają z dwóch stron linijki, a w lewym dolnym rogu znajduje się globalny układ współrzędnych. Opcje widoku tego okna można modyfikować za pomocą polecenia *Ustawienia graficzne* (*Graphics Setup*) wywoływanego poprzez ikonę zaznaczoną na rys. 3.27 lub skrót klawiszowy Ctrl+D. Jeżeli którekolwiek z prezentowanych ikon nie są widoczne, należy kliknąć prawym klawiszem myszy w miejscu przewidzianym na paski narzędziowe, a następnie aktywować odpowiednie ikony, potrzebne w danej chwili.

Opcje prezentacji zawartości okna graficznego są dostępne na dwa sposoby: w postaci ikon (pasek narzędziowy znajdujący się zwykle u góry okna) oraz w postaci rozwijalnego drzewa (zwykle po lewej stronie okna graficznego).

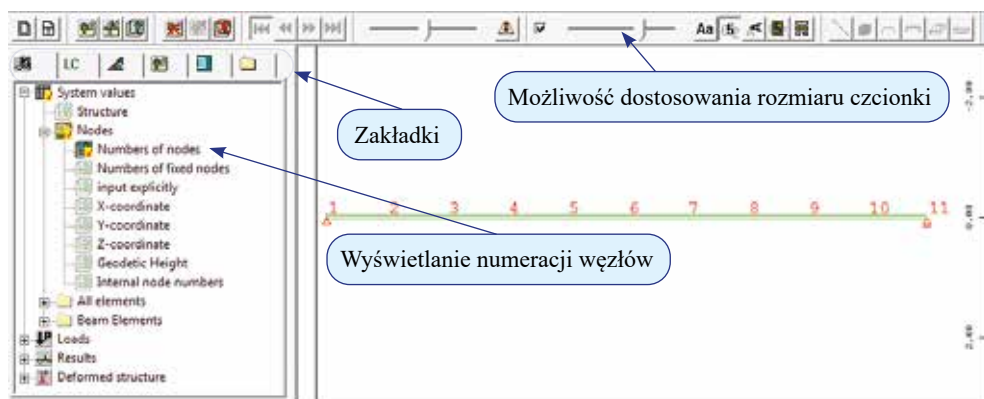
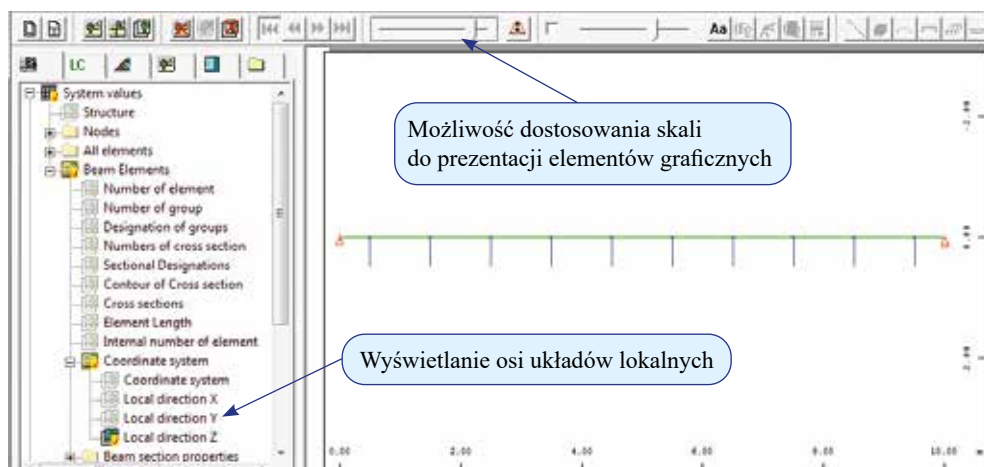
Dzielią się one na następujące grupy:

- *Wartości systemowe* (*System values*) – opcje dotyczące modelu układu;
- *Obciążenia* (*Loads*);
- *Wyniki* (*Results*);
- *Wartości projektowe* (*Design Values*), w analizowanym przykładzie nieaktywne;
- *Deformacja układu* (*Deformed structure*), tylko poprzez drzewo.

W dalszej części do prezentacji wyników zostanie wykorzystane rozwijalne drzewo, które jest wygodniejsze w obsłudze. Należy jednak pamiętać, że drzewo nie ma wszystkich możliwych opcji dostępnych za pomocą ikon (jak np. ustawienia skali widoku czy ograniczenia prezentowanych wartości tylko do wybranych węzłów czy elementów). W górnej części rozwijalnego drzewa (rys. 3.28) znajduje się sześć zakładek, które w kolejności od lewej odpowiadają za następujący wybór: (1) opcji prezentacji, (2) przypadku obciążenia, (3) prezentacji (tekst, wektor, wykres, mapa), (4) warstwy rysunku, (5) numeracji, (6) bazy danych.

Jako przykładowy wynik z grupy *Wartości systemowe* (*System values*) wybrano prezentację numeracji węzłów (rys. 3.28). W kolejnym wyniku z tej grupy zademonstrowano osie z lokalnego układu współrzędnych elementów (rys. 3.29). Warto zauważyć, że przyciski aktywnych opcji są wyróżnione kolorowym podświetleniem.

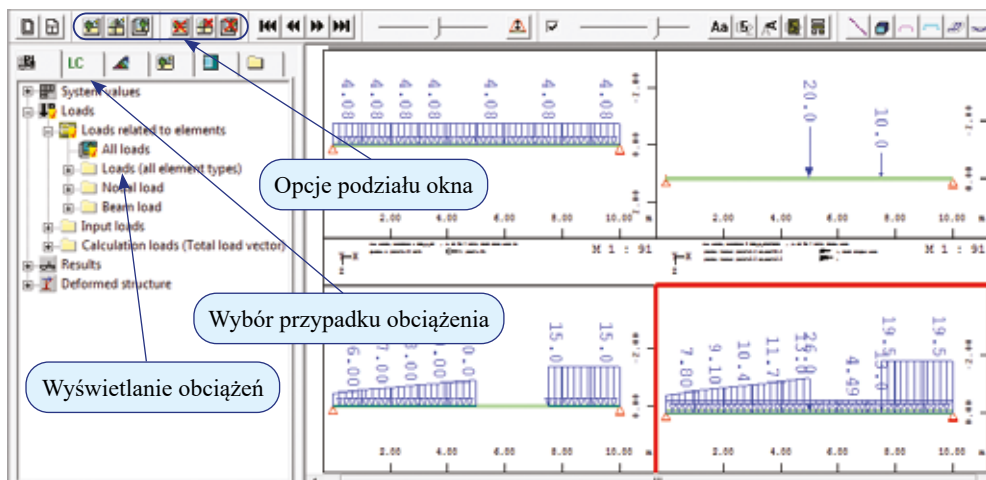
Kolejna grupa *Obciążenia* (*Loads*) pozwala na wizualne sprawdzenie, czy obciążenie układu zostało zdefiniowane poprawnie. W tym zadaniu okno graficzne jest podzielone na cztery części, do których przypisano różne przypadki obciążenia (rys. 3.30). Zabieg podziału okna można wykonać za pomocą ustawień dostępnych w *Graphics Setup* (Ctrl+D). Dodatkowo, aby uzupełnić pola podziału okna, niezbędne jest kliknięcie ikony *New Picture with One layer* (druga ikona od lewej w opcjach podziału okna widocznych na rys. 3.30). Zdefiniowanie ustawień obrazu wymaga w pierwszej kolejności uaktywnienia pożądanego pola poprzez kliknięcie w jego obszarze. Aktywne pole jest zaznaczone czerwoną obwódką (rys. 3.30, pole dolne z lewej strony). Następnie należy przełączyć zakładkę z (1) opcji prezentacji na (2) przypadki obciążenia, aby zmienić prezentowany schemat obciążenia. Czynności te należy wykonać trzykrotnie, każdorazowo zmieniając przypadek obciążenia na kolejny.

Rys. 3.28. Moduł *WinGraf* – numeracja węzłówRys. 3.29. Moduł *WinGraf* – lokalne układy współrzędnych, osie z poszczególnych elementów skończonych, tu skierowane do dołu

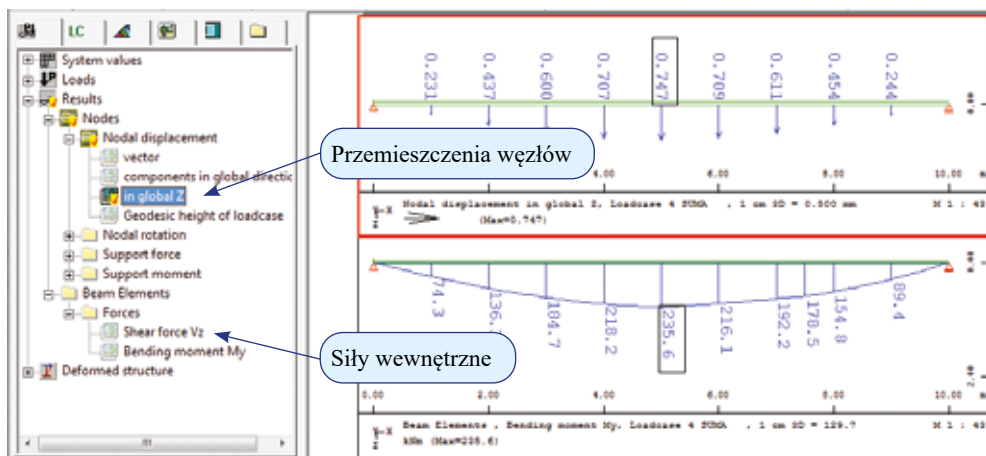
Wyniki obliczeń są dostępne poprzez grupę *Rezultaty (Results)*. Jako przykład na rys. 3.31 przedstawiono przemieszczenia węzłów w kierunku globalnej osi Z oraz wykres momentów zginających M_y w przypadku obciążenia nr 4 (wartości w ramkach wskazują ekstrema). W opcjach wyboru pojawiają się pewne pozorne ograniczenia, np. brakuje możliwości podglądu przemieszczeń na kierunkach X oraz Y , mimo że wcześniej formalnie zdefiniowano przestrzeń zanurzenia układu jako 3D. Dzieje się tak, ponieważ moduł *WinGraf* w sposób inteligentny nie proponuje podglądu wyników, w których rozwiązaniem są wartości zerowe (zdefiniowane obciążenia nie wywołują przemieszczeń na tych kierunkach).

UWAGA. Przy interpretacji i ocenie wyników należy pamiętać o kilku kluczowych zasadach ustalonych przez twórców programu:

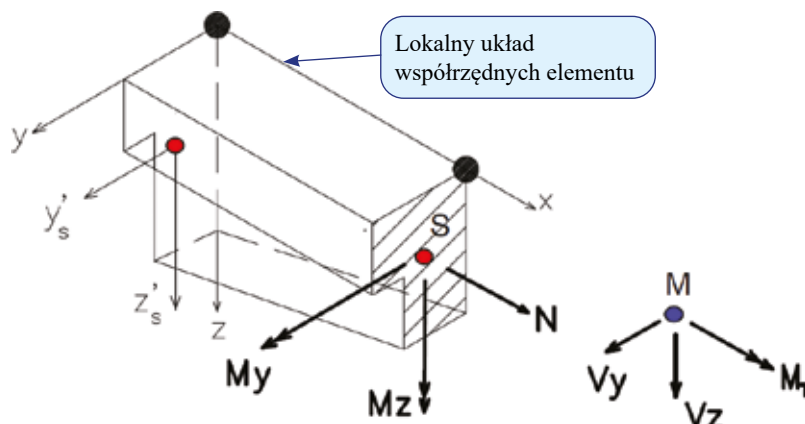
- wyniki dotyczące węzłów układu (przeszczenia, reakcje) prezentowane są zawsze w odniesieniu do globalnego układu współrzędnych;
- wyniki dotyczące elementów (siły wewnętrzne, naprężenia) prezentowane są zawsze w odniesieniu do ich lokalnego układu współrzędnych;
- konwencja znakowania sił wewnętrznych w różnych programach MES może być inna, stąd należy zawsze sprawdzić ustalone w danym systemie zasady przed przystąpieniem do ich interpretacji. W programie SOFiSTiK obowiązuje konwencja przedstawiona na rys. 3.32.



Rys. 3.30. Moduł *WinGraf* – wyświetlenie obciążeń



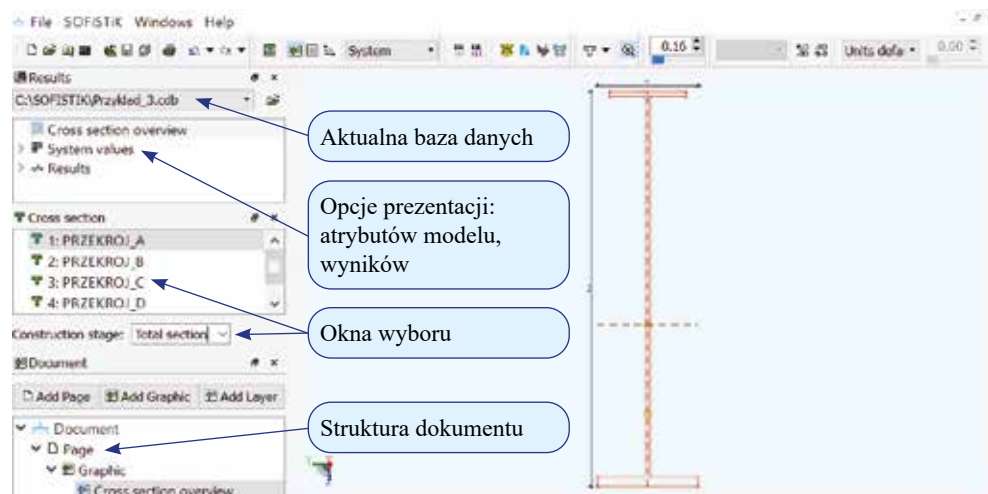
Rys. 3.31. Moduł *WinGraf* – przeszczenia węzłów oraz wykres momentów zginających wraz z opisem jednostek i ekstremów



Rys. 3.32. Pomoc/ instrukcja do programu *SOFIMSHA*
 – konwencja znakowania sił wewnętrznych w elementach prętowych, układ osi y i z
 wynika z definicji charakterystyk przekrojowych w module *AQUA*

Tabelaryczna prezentacja wyników

Zestawienie wyników w sposób tabelaryczny można uzyskać w module *Result Viewer*. Ekran programu oraz jego podstawowe funkcje zawarto na rys. 3.33. Aby otrzymać określoną tabelę, wykorzystuje się rozwijalne drzewko w oknie prezentacji. Na rys. 3.34 jako przykład z grupy *Wartości systemowe (System values)* pokazano tabelę współrzędnych X węzłów belki, natomiast na rys. 3.35 jako przykład z grupy *Rezultaty (Results)* przedstawiono przemieszczenia węzłów w kierunku globalnej osi Z dla przypadku obciążenia nr 4.



Rys. 3.33. Moduł *Result Viewer*

NR	X [m]
1	0,000
2	1,000
3	2,000
4	3,000
5	4,000
6	5,000
7	6,000
8	7,000
9	8,000
10	9,000

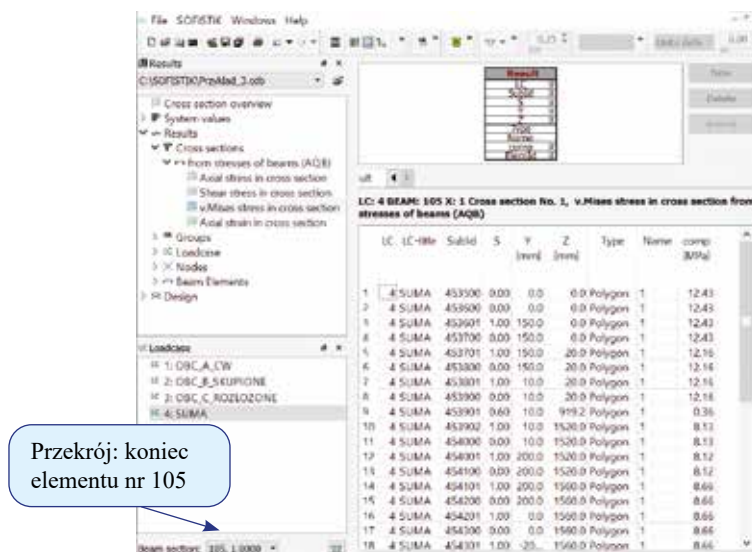
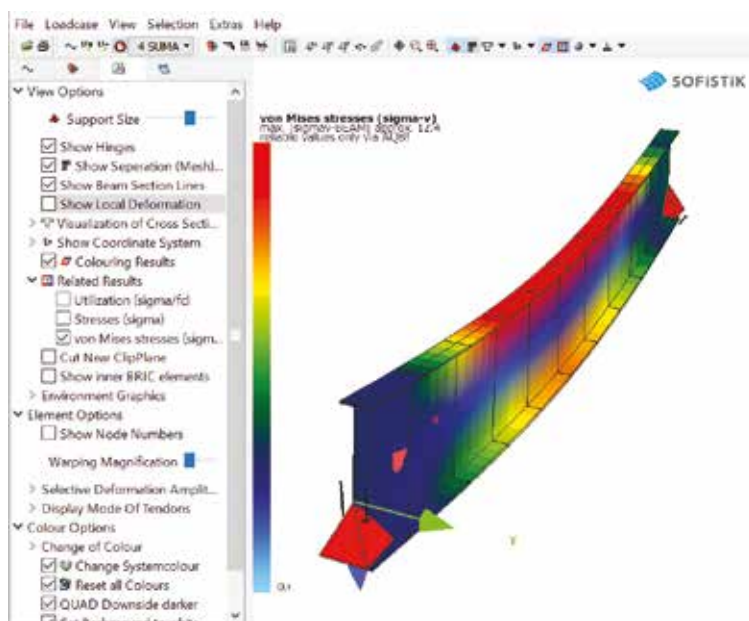
Rys. 3.34. Moduł *Result Viewer* – współrzędne X węzłów belki

LC	LC-title	Number	u-Z [mm]
1	4 SUMA	1	0,000
2	4 SUMA	2	0,231
3	4 SUMA	3	0,437
4	4 SUMA	4	0,600
5	4 SUMA	5	0,707
6	4 SUMA	6	0,747
7	4 SUMA	7	0,709
8	4 SUMA	8	0,611
9	4 SUMA	9	0,454
10	4 SUMA	10	0,244

Rys. 3.35. Moduł *Result Viewer* – przemieszczenia węzłów w kierunku globalnej osi Z

3.5. Prezentacja wyników – naprężenia

Po wykonaniu obliczeń programem *ASE* miejscem, w którym można uzyskać wartości naprężenia w elementach, jest moduł *Result Viewer* (rys. 3.36), ewentualnie ich przybliżoną wizualizację można wygenerować w module *Animator* (rys. 3.37). Bardziej czytelna prezentacja stanu naprężenia w przypadku elementów prętowych w programie *SOFiSTiK* wymaga dodatkowych zabiegów. Wynika to z konieczności uwzględnienia odpowiednich hipotez dotyczących ich rozkładu w przekroju w ramach odpowiedniej teorii prętów, omawianych np. na kursie wytrzymałości materiałów czy kursie teorii prętów cienkościennych.

Rys. 3.36. Moduł *Result Viewer* – naprężenia według hipotezy HMMRys. 3.37. Moduł *Animator* – naprężenia według hipotezy HMM

Prezentacja wartości naprężenia w postaci wykresów w module *WinGraf* wymaga uaktywnienia grupy *Wartości Projektowe (Design Values)* (zob. rys. 3.38) oraz wywołania/użycia w kodzie danych dodatkowego programu *AQB*. Program *AQB* służy do wymiarowania elementów prętowych. W podstawowej wersji można go wykorzystać w zakresie sprawdze-

nia stanu naprężenia. Poniższy zapis należy umieścić w kodzie pod programem *ASE*, w którym wykonuje się obliczenia statyczne dla poszczególnych przypadków obciążenia.

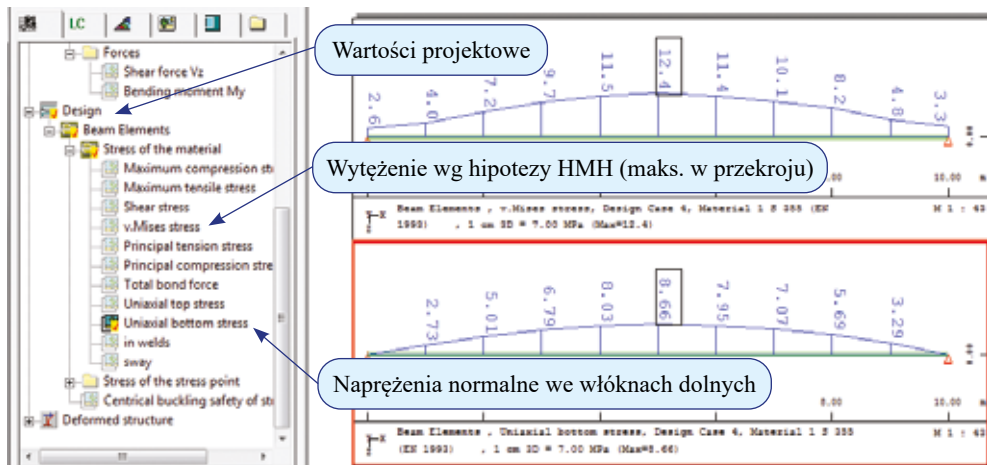
```
+prog aqb
head naprezenia
lc (1 4 1)

stre e

end
```

Obliczenia dla przypadków obciążenia od 1 do 4 (nawiasy wskazują na użycie generatora liniowego) wpisywane w programie *AQB* po komendzie *lc*. Funkcja *stre* określa zakres obliczeń. Opcja *e* wybiera tylko sprawdzenie stanu naprężenia.

Przykładowe wyniki (rys. 3.38) przedstawiają wykres zmiany wartości maksymalnego wyężenia wg hipotezy HMM w kolejnych przekrojach belki oraz średnie wartości naprężenia normalnego we włóknach dolnych przekroju.



Rys. 3.38. Moduł *WinGraf* – ekstremalne wyężenia według hipotezy HMM oraz naprężenia normalne we włóknach dolnych

3.6. Wykorzystanie zmiennych oraz pętli – obciążenia ruchome

Jak wspomniano wcześniej, wprowadzanie danych do programu SOFiSTiK w postaci odpowiedniego kodu jest tzw. językiem problemowo zorientowanym, interpretowanym przez system. Jedną z zalet takiego podejścia jest możliwość użycia specyficznego zapisu zmiennych oraz pętli charakterystycznych w procesie programowania. Dzięki temu możliwe są skracanie zapisu, parametryzacja obiektów typowych czy też generowanie obciążenia ruchomego (np. pojazdu bądź budowa linii i powierzchni wpływowych), co zostanie pokazane w niniejszym rozdziale.

Przykład 3.6. Zdefiniować przejazd siły skupionej o wartości 1 kN z krokiem 0,5 m przez belkę z podziałem na dziesięć elementów skończonych, stworzoną w przykładzie 3.2.2 (w rozdziale 3.2). Potraktować obciążenie jako quasistatyczne z pominięciem efektów dynamicznych w postaci sekwencji kolejnych „skokowych” ustawień siły.

Do zdefiniowania obciążenia zostanie użyty program *SOFILOAD* z odpowiednio wbudowaną pętlą. Aby krok położenia (przejazdu) siły nie zależał od geometrii/ rozstawu węzłów, wykorzystany będzie wariant obciążenia typu wolnego (*poін*).

```
+prog sofiload
head przejazd_sily
let#lc 101

let#poz_x 0

loop 21

  lc no #lc titl przejazd

  poin ref bgrp type pzz p 1 x #poz_x y 0 z 0

  let#lc #lc+1

  let#poz_x #poz_x+0.5

endloop
end
```

Wywołanie programu *SOFILOAD*.

Każde położenie siły ma być generowane jako osobny przypadek obciążenia. W tym celu za pomocą komendy *let* wprowadza się zmienną numeryczną (tutaj o nazwie *#lc*). Nazwę zmiennej otwiera symbol *#*, po którym występuje dowolny ciąg znaków bez spacji. Ważne, aby pomiędzy komendą *let* a nazwą zmiennej nie stosować spacji. Tutaj jako wyjściową wartość zmiennej *#lc* przyjęto liczbę 101.

Kolejna zmienna *#poz_x* definiuje położenie siły. Jako wartość początkową przyjęto 0 (skraj belki). Komendą *loop* ustala się początek pętli. Po komendzie podaje się przewidzianą liczbę powtórzeń, tutaj 21, co przy kroku 0,5 m odpowiada końcowi belki.

Komendą *lc* (tak jak poprzednio) tworzy się przypadek obciążenia, tutaj o numerze określonym zmienną *#lc* (o wartości początkowej 101). W celu zachowania czytelności zapisu zaleca się komendy wewnątrz pętli odsunąć od lewej krawędzi o kilka spacji (tu dwie).

Definicja obciążenia siłą skupioną, typu wolnego (*poін*) w układzie globalnym w kierunku osi *Z*. Pozycję siły na osi *X* określa zmienna *#poz_x*, współrzędne *Y* oraz *Z* są stałe o wartości 0.

Rekurencyjna aktualizacja numeru *#lc* przypadku obciążenia o wartość 1. Po przejściu pętli numery te będą tworzyć ciąg {101, 102, 103, ..., 121}.

Rekurencyjna aktualizacja wartości zmiennej *#poz_x* odpowiedzialnej za pozycję siły skupionej. Po przejściu pętli położenia te będą tworzyć ciąg: {0; 0,5; 1,0; ...; 10} m.

Komenda kończąca pętlę.

Obliczenia wygenerowanego powyżej szeregu quasistatycznych przypadków obciążenia przeprowadza się z wykorzystaniem modułu *ASE*.

```
+prog ase
head przejazd_obliczenia
lc (101 121 1)
end
```

Aby zakończyć przykład, należy zarówno program *SOFILOAD*, jak i *ASE* przeliczyć. Wykonuje się to w module *WPS*.

Wykorzystując moduł *Result Viewer*, na rys. 3.39 przedstawiono przykładowe wyniki analizy w postaci tabelarycznego zestawienia wartości momentów zginających na końcu elementu nr 105 (środek rozpiętości belki), wywołanych poruszającą się siłą, zapisanych w przypadkach obciążenia {101, 102, ..., 121} (należy je wybrać w oknie *Przypadki obciążenia (Loadcase)*). Selektywny wybór wielkości do prezentacji (nr elementu skończonego, lokalizacja przekroju poprzecznego) wymaga użycia odpowiedniego tzw. filtra w postaci ograniczeń w kolumnach *NR* oraz *X [m]*. Filtr uruchamia się, klikając prawym klawiszem w wybrany nagłówek tabeli. Przykładowo, w kolumnie *NR* kolejne kroki wyboru powinny mieć następującą sekwencję: *Filter* → *Range* → *Filter settings: equal* → *105*, następnie w kolumnie *X [m]*: *Filter* → *Range* → *Filter settings: equal* → *1[m]*.

Zaprezentowany wynik można wykorzystać do stworzenia klasycznych linii wpływo- wych. Wystarczy kolumnę *MY kNm* z tabeli skopiować do dowolnego programu graficznego (np. *Sigma Plot*, *Grapher*, *Excel* itp.), a następnie dorobić ręcznie kolumnę współrzędnej globalnej odpowiadającej położeniu siły.

The screenshot shows the 'Results' window of a software application. On the left, a tree view shows the project structure, with 'Bending moment My' selected under 'Forces'. The main area displays a table titled 'LC: 101-121, Beam Elements Bending moment My'. A blue callout box labeled 'Filtry' points to the 'NR' and 'X [m]' columns. The table contains 21 rows of data, all filtered to NR=105 and X=1.000.

LC	LC-title	NR	X [m]	MY kNm
1	101 PRZEJAZD	105	1.000	0.00
2	102 PRZEJAZD	105	1.000	0.25
3	103 PRZEJAZD	105	1.000	0.50
4	104 PRZEJAZD	105	1.000	0.75
5	105 PRZEJAZD	105	1.000	1.00
6	106 PRZEJAZD	105	1.000	1.25
7	107 PRZEJAZD	105	1.000	1.50
8	108 PRZEJAZD	105	1.000	1.75
9	109 PRZEJAZD	105	1.000	2.00
10	110 PRZEJAZD	105	1.000	2.25
11	111 PRZEJAZD	105	1.000	2.50
12	112 PRZEJAZD	105	1.000	2.25
13	113 PRZEJAZD	105	1.000	2.00
14	114 PRZEJAZD	105	1.000	1.75
15	115 PRZEJAZD	105	1.000	1.50
16	116 PRZEJAZD	105	1.000	1.25
17	117 PRZEJAZD	105	1.000	1.00
18	118 PRZEJAZD	105	1.000	0.75
19	119 PRZEJAZD	105	1.000	0.50
20	120 PRZEJAZD	105	1.000	0.25
21	121 PRZEJAZD	105	1.000	0.00

Rys. 3.39. Moduł *Result Viewer* – momenty zginające na końcu elementu nr 105 od obciążenia przejazdem

3.7. Kombinacje obciążeń i obwiednie

Najprostszy sposób tworzenia kombinacji obciążeń przedstawiono w przykładzie 3.3d (rozdział 3.3). Polega on na sumowaniu w programie *ASE* wygenerowanych wcześniej przypadków za pomocą komendy *lcc* z odpowiednim mnożnikiem. Analogiczne narzędzie występuje również w programie *SOFILOAD* – jest to komenda *copy*.

W niniejszym rozdziale zajmiemy się technikami tworzenia bardziej rozbudowanych wersji kombinacji obciążeń oraz poszukiwania wielości ekstremów na drodze generowania obwiedni. Do tego celu zostanie wykorzystany program *MAXIMA*. Podejście to bardzo dobrze sprawdza się w przypadku dużej liczby wariantów obciążeń i jest wygodnym sposobem tworzenia tzw. kombinacji normowych.

Przykład 3.7. Wyznaczyć obwiednie maksymalnych momentów zginających, ekstremalnych sił tnących oraz maksymalnych przemieszczeń pionowych przy uwzględnieniu odpowiednie kombinacje z następujących wcześniej wygenerowanych przypadków obciążeń:

- *LC 1* – cw – obciążenie stałe (przykład 3.3a);
- *LC 2* – siły skupione – obciążenie eksploatacyjne (przykład 3.3b);
- *LC 3* – siły rozłożone – obciążenie eksploatacyjne (przykład 3.3c);
 - *LC 101 ÷ LC 121* – przejazd pojazdu (przykład 3.6) – obciążenia eksploatacyjne, a więc na konstrukcji w danym czasie może się znajdować tylko jedno z szeregu ustawień o numerach {101, 102, ..., 121}.

W przypadku obwiedni sił wewnętrznych wprowadzić następujące mnożniki: 1,1 dla obciążenia stałego, 1,3 dla obciążeń eksploatacyjnych.

```
+prog maxima
head kombinacje+obwiednie
ctrl save 1
```

```
comb no 1 extr stan base 1000 titl sgn
```

```
lc 1 type g fact 1.1
```

Przy wyborze wartości funkcji *save* równej 1 program zapisze/ zarchiwuje uzyskane wyniki na dysku.

Komenda *comb* ustala regułę kombinacji. Każda reguła musi posiadać niepowtarzalny numer, tu 1. Funkcja *extr* określa rodzaj superpozycji. Parametr *stan* tworzy standardową kombinację bez domyślnych mnożników. Funkcja *base* definiuje numer bazowy przypadku obciążenia (ze względu na tworzenie nr. szczegółowych (rys. 3.41) musi być to wielokrotność liczby 100, maks. 9900; tutaj przyjęto 1000).

Po komendzie *lc* (rys. 3.40) podaje się numery zdefiniowanych wcześniej przypadków obciążenia wchodzących w skład tworzonej reguły (tu nr 1). Za pomocą funkcji *type* określa się sposób traktowania tego obciążenia. Parametr *g* oznacza, że przypadek ten zawsze będzie składnikiem kombinacji (obciążenie stałe). Funkcja *fact* określa mnożnik, w rozważanym przypadku obciążenia równy 1,1.

`lc 2,3 type q fact 1.3`

Dla przypadków obciążenia nr 2 i 3 *type* zdefiniowano jako *q*. Parametr *q* (rys. 3.40) oznacza, że ujęcie tych wariantów w kombinacji będzie warunkowe, tzn. tylko wówczas, gdy wywołają one przyrost poszukiwanego ekstremum. Mnożnik dla tych przypadków określono jako 1,3.

`lc (101 121 1) type a1 fact 1.3`

Szereg przypadków {101, 102, ..., 121} określa ruch pojazdu. Dla funkcji *type* parametr *a1* (rys. 3.40) wskazuje, że do kombinacji będzie wzięte tylko jedno ustawienie – to, które wywołuje największy przyrost poszukiwanego ekstremum. Mnożnik dla tych przypadków również określono jako 1,3.

LC			
Item	Description	Unit	Default
NO	0 All load cases of the defined action are used	-	*
	-1 All load cases of the defined category are used		
TYPE	Type of the load	LIT	*
	G Always (permanent) safety factor action-wise or PERM from record ACT item SUP		
	PERC Always (permanent), however, with variable factors safety factor load-case-wise from record ACT item SUP		
	Q Conditional (only unfavourable) (P has the same meaning) or		
	A1 Alternative load case group A1, exclusive within one action		
	...		
	A99 Alternative load case group A99 or EXCL from record ACT item SUP		
	X1 Alternative load case group X1 with changing sign		
	...		
	X99 Alternative load case group X99 with changing sign or USEX from record ACT item SUP		
	Default: see below For non-linear load cases only AG1 is permitted.		
FACT	Factor with which the corresponding internal force of the initial load case is multiplied	-/LIT	1.0

Typ obciążenia: stałe, zawsze brane do kombinacji

Typ obciążenia: warunkowe, brane do kombinacji, gdy powoduje wzrost ekstremum

Typ obciążenia: alternatywne, brane do kombinacji, gdy powoduje wzrost ekstremum; poszczególne przypadki będące w jednej grupie, np. a1, wykluczają się.

Rys. 3.40. Pomoc/ instrukcja do programu MAXIMA – wybrane funkcje komendy *lc*

```
comb no 2 extr stan base 1100 titl sgu
```

Regułę nr 2 kombinacji obciążeń tworzy się kolejną komendą *comb*. Różnica dotyczy tylko mnożników, a w istocie ich braku. Numer bazowy tworzących przypadków wynosi 1100.

```
lc 1 type g fact 1
```

Powtórzenie zapisu przypadków wchodzących w skład reguły nr 2; jedyną różnicę stanowią wartości mnożników, które w każdym przypadku są równe 1.

```
lc 2,3 type q fact 1
```

```
lc (101 121 1) type a1 fact 1
```

```
supp comb 1 extr mami etyp beam type vz
```

Za pomocą komendy *supp* definiuje się poszukiwaną obwiednię. W pierwszej kolejności za pomocą funkcji *comb* wybiera się nr reguły kombinacji, tu 1, następnie zaś typ poszukiwanego ekstremum; dla funkcji *extr* parametr *mami* oznacza, że mają być określone wartości zarówno *max*, jak i *min*. Funkcją *etyp* ustala się, czy poszukiwana wielkość jest węzłowa, elementowa itd.; w tym przypadku – elementowa (parametr *beam*). Na koniec z tabeli dostępnych wielkości dla danego *etyp* (rys. 3.41), definiuje się rodzaj poszukiwanej obwiedni. Tutaj wybrano siłę tnącą (parametr *Vz*).

• for beams - ETYP BEAM and for external beam sections - ETYP BSCT

TYPE	Designation	Load case number			max/min
		3	2	2	
		D	D	D	
N	Beam normal force	*	*		21/22
VY	Beam shear force Vy	*	*		23/24
VZ	Beam shear force Vz	*	*	*	25/26
MT	Torsional moment Mt	*	*	*	27/28
MY	Bending moment My	*	*	*	29/30
MZ	Bending moment Mz	*	*	*	31/32
MB	Warping moment	*	*	*	33/34
MT2	Secondary torsional moment	*	*	*	35/36
UX	Displacement u-x	*	*	*	71/72
UY	Displacement u-y	*	*	*	73/74
UZ	Displacement u-z	*	*	*	75/76
URX	Rotation phi-x	*	*	*	77/78
URY	Rotation phi-y	*	*	*	79/80

Nr (+) przypadku, pod którym figurować będzie obwiednia po dodaniu do nr. bazowego 1000, np. dla momentu My będą to: max LC 1029, min LC 1030.

Siły wewnętrzne

Rys. 3.41. Pomoc/ instrukcja do programu *MAXIMA* – wybrane wartości dla funkcji *etyp beam* komendy *supp*

• for nodes - ETYP NODE

TYPE	Designation	3 D	2 D	2 D S L	Load case number max/min
PX	Support reaction in X direction	*	*		51/52
PY	Support reaction in Y direction	*	*		53/54
PZ	Support reaction in Z direction	*	*	*	55/56
P	all support reactions	*	*	*	
MX	Support moment about X direction	*	*	*	57/58
MY	Support moment about Y direction	*	*	*	59/60
MZ	Support moment about Z direction	*	*	*	61/62
MB	Support warping moment	*	*	*	91/92
UX	Nodal displacement in X direction	*	*	*	71/72
UY	Nodal displacement in Y direction	*	*	*	73/74
UZ	Nodal displacement in Z direction	*	*	*	75/76
URX	Nodal rotation about X direction	*	*	*	77/78
URY	Nodal rotation about Y direction	*	*	*	79/80
URZ	Nodal rotation about Z direction	*	*	*	81/82
URB	Nodal rotation due to warping moment	*	*	*	93/94

Reakcje

Przemieszczenia, obroty

Rys. 3.42. Pomoc/ instrukcja do programu *MAXIMA* – wybrane wartości dla funkcji *etyp node* komendy *supp*

```
supp comb 1 extr max etyp beam type my
```

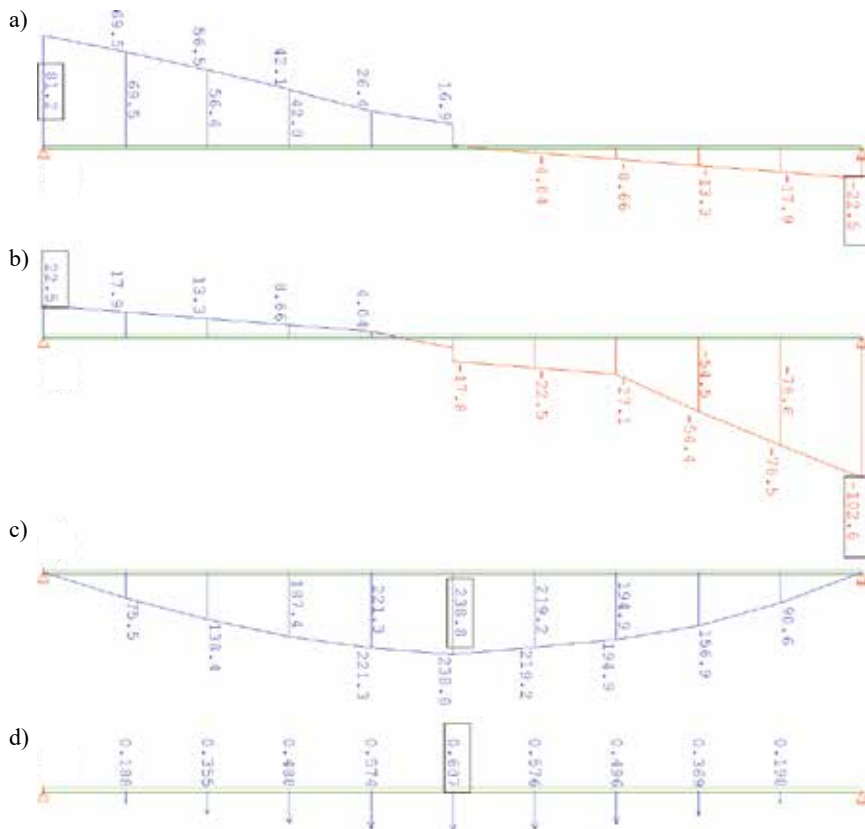
Kolejna poszukiwana obwiednia zdefiniowana jest jako elementowa (parametr *beam*). Poszukuje się wartości maksymalnych (parametr *max*) momentu *My* (parametr *my*). Nadal powstaje ona z reguły kombinacji (*comb*) nr 1.

```
supp comb 2 extr max etyp node type uz
```

Ostatnia poszukiwana obwiednia jest zdefiniowana jako węzłowa (parametr *node*). Poszukuje się wartości maksymalnych (parametr *max*) przemieszczeń pionowych (parametr *uz*). Tym razem powstaje ona z reguły (*comb*) kombinacji nr 2. Inne wartości funkcji *etyp node* zawiera rys. 3.42.

```
end
```

Wyniki utworzone w module *WinGraf* w postaci wykresów obwiedni sił tnących, momentów oraz przemieszczeń węzłów przedstawiono na rys. 3.43. Ważne, aby przy wyborze wykresu do prezentacji w module *WinGraf* w pierwszej kolejności określić rodzaj zmiennej, np. moment zginający *My*, a dopiero później przypadek, np. 1029: *MAX-MY BEAM*. Przy zastosowaniu odwrotnej kolejności przypadki zawierające obwiednie mogą nie być widoczne.



Rys. 3.43. Obwiednie: maksimum sił tnących [kN] (a), minimum sił tnących [kN] (b), maksimum momentów zginających [kNm] (c) oraz maksimum przemieszczeń pionowych [mm] (d)

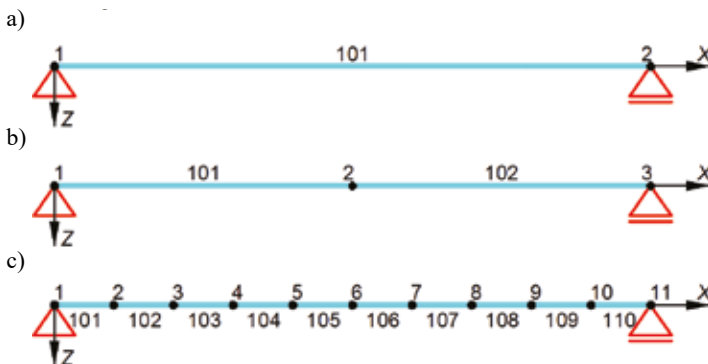
3.8. Uwagi o zbieżności rozwiązań MES, udoskonalenie typu h

W dotychczasowych rozważaniach, skoncentrowanych na sposobie korzystania z programu SOFiSTiK, jedynie zasygnalizowano potrzebę świadomego wyboru typu elementu skończonego oraz poziomu dokładności dyskretyzacji (gęstości siatki podziału na elementy). Korzystając z programów MES, nigdy nie wolno zapominać, że **metoda elementów skończonych jest metodą przybliżoną!** Stąd, jak w każdej metodzie numerycznej, tak i w MES poprawność i dokładność uzyskanych wyników zależą od spełnienia kryteriów zbieżności stosowanej metody. Z perspektywy użytkownika komercyjnych programów MES najczęściej sprowadza się to do wyboru odpowiedniego elementu skończonego i właściwego doboru siatki dyskretyzacyjnej. Problematykę tę zasygnalizujemy w niniejszym rozdziale. Program SOFiSTiK oferuje stosunkowo ubogą bibliotekę elementów skończonych, tak że przedstawieniu ich podstaw teoretycznych nie poświęca się zbyt wiele uwagi. Dlatego przed przystąpieniem do właściwych obliczeń konstrukcji warto zbadać ich możliwości i zachowanie na własnych testach, które wskazałyby na zasadność ich użycia do konkretnych zagadnień inżynierskich.

W prezentowanych zadaniach wykorzystano jednowymiarowe, dwuwęzłowe przestrzenne prętowe/ belkowe elementy skończone typu Timoszenki, uwzględniające efekt ścinania o sześciu stopniach swobody w węźle (trzy translacje/ przesunięcia: $u-X$, $u-Y$, $u-Z$ oraz trzy parametry rotacyjne/ obroty: $\phi-X$, $\phi-Y$, $\phi-Z$). Dodatkowo uwzględniono w nich mimośród osi belki.

Zaznaczmy, że w klasycznym sformułowaniu elementy belkowe typu Timoszenki są klasy C^0 o liniowych funkcjach kształtu w wariacie dwuwęzłowym. Niestety te dwuwęzłowe elementy o liniowych funkcjach kształtu przy stosowaniu tzw. całkowania pełnego (FI) wykazują bardzo silny negatywny efekt numeryczny, tzw. blokadę/ zakleszczenie związane ze ścinaniem poprzecznym (*shear locking*) w przypadku belek smukłych (zob. np. COOK I IN. [2002]). Problemowi blokady, który występuje w elementach skończonych niskiego rzędu, nie tylko belkowych (zob. np. CHRÓŚCIELEWSKI I IN. [2004]), poświęcono wiele uwagi, co znalazło odzwierciedlenie w bogatej literaturze na ten temat. W efekcie badań prowadzonych w wielu ośrodkach naukowych na świecie wypracowano szereg różnych metod redukcji blokady o różnym poziomie skuteczności. I tak, analogicznie jak przy wyprowadzeniu wzorów transformacyjnych metody przemieszczeń belki Bernoulliego-Eulera, w sformułowaniu elementu belkowego dostępnego w programie SOFiSTiK od funkcji aproksymujących ugięcia i obroty żąda się ścisłego spełnienia jednorodnych równań różniczkowych belki Timoszenki (HARTMANN I KATZ [2002], [2007]). W wyniku tego warunku otrzymano dwuwęzłowy element skończony wolny od blokady, w którym funkcja ugięcia jest wyrażona przez wielomian trzeciego stopnia, natomiast funkcja obrotów – przez wielomian drugiego stopnia. W różnych systemach obliczeniowych w odniesieniu do elementów prętowych/ belkowych niskiego rzędu mogą być stosowane inne techniki/ metody redukcji blokady/ zakleszczenia. Sprawdzenie, nawet obliczeniowe, tego typu informacji i jednocześnie „nabranie zaufania” do użytego elementu w momencie rozpoczęcia pracy z nowym oprogramowaniem powinno mieć kluczowe znaczenie dla użytkownika.

Typ elementu skończonego wykorzystywanego w obliczeniach implikuje potrzebę stosowania odpowiedniego poziomu dokładności dyskretyzacji. Przykładem niech będzie badanie rozwiązania rozważanej wcześniej belki (przykład 3.2) przy zastosowaniu dyskretyzacji z użyciem: jednego (a), dwóch (b) oraz dziesięciu (c) elementów skończonych (rys. 3.44).



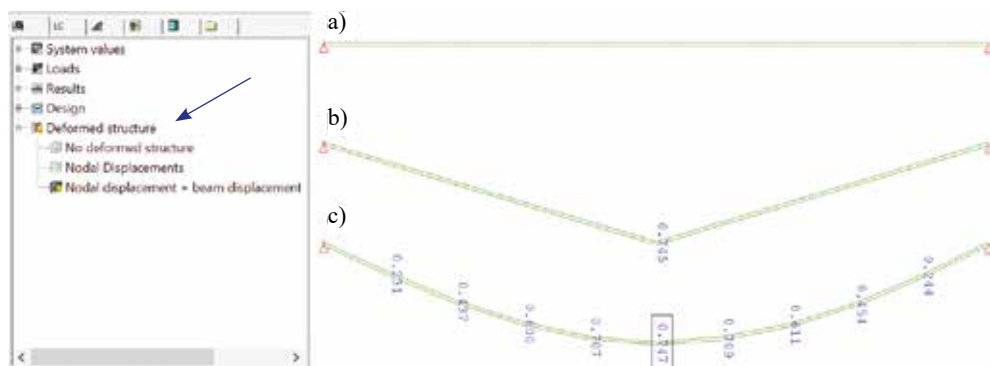
Rys. 3.44. Dyskretyzacje belki swobodnie podpartej utworzone z: jednego (a), dwóch (b) oraz dziesięciu (c) elementów skończonych

Wszystkie inne parametry nie ulegały zmianie. Układ obciążono, wykorzystując przypadek nr 4 (przykład 3.3d z rozdziału 3.3). Oczywiście obciążenia typu strukturalnego, powiązane sztywno z konkretną siatką podziału, zostały zmienione na obciążenia typu wolnego, niezależnego od dyskretyzacji.

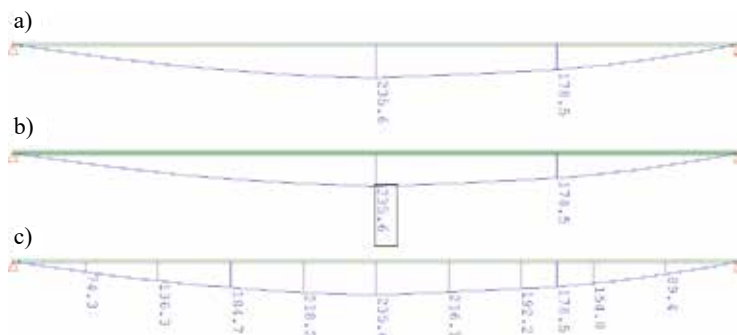
Jako pierwszy wynik zaprezentowano belki z włączoną deformacją i z podanymi wartościami pionowych przemieszczeń w węzłach (rys. 3.45). Wizualizacja deformacji w dwóch pierwszych przypadkach znacznie odbiega od oczekiwanej; wizualizacja ta bazuje niestety tylko na krzywej łamanej przeprowadzanej przez węzły.

Więcej informacji o charakterze funkcji aproksymacyjnych/ kształtu stosowanych w elementach skończonych można ewentualnie uzyskać, analizując zmienność sił wewnętrznych w obszarze elementu, o ile nie podlegały one po obliczeniu dalszej obróbce graficznej. Na rys. 3.46 pokazano wizualizację przebiegu momentów zginających. Okazuje się, że wynik sił wewnętrznych nie jest mocno zależny od podziału na elementy.

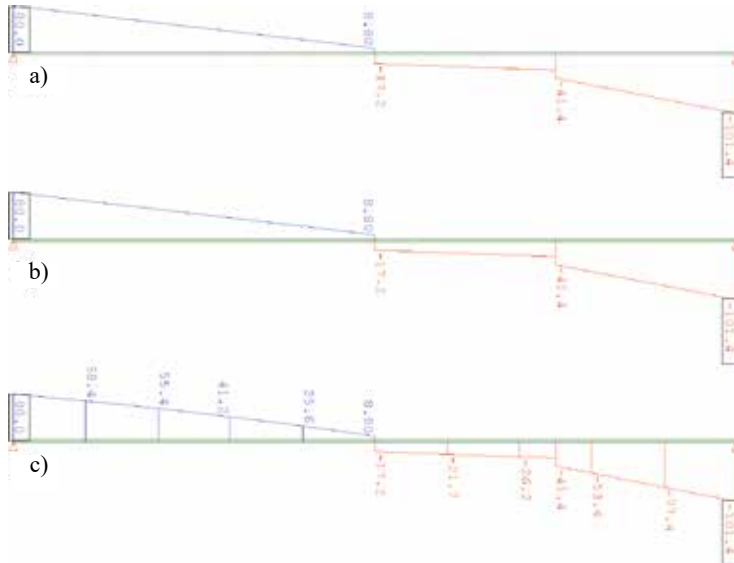
W przypadku stosowania modeli z niskim poziomem dyskretyzacji problem może się pojawić tylko wtedy, gdy chce się odczytać wartość siły w przekroju dowolnym, tj. innym niż ekstremalny. Ostatni przedstawiony wynik to wykresy ekstremalnego naprężenia w przekroju wg hipotezy HMH (rys. 3.48). W tym przypadku również, aby możliwe było uzyskanie poprawnego rozwiązania, niezbędna jest bogatsza dyskretyzacja.



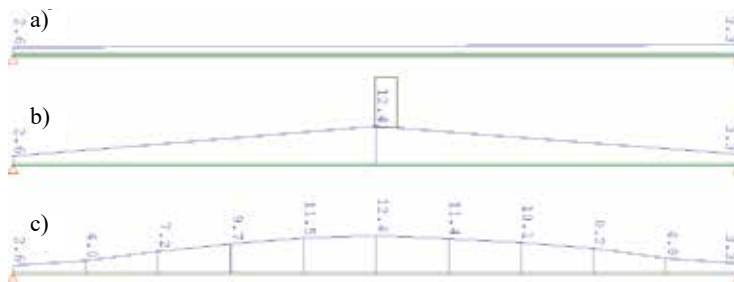
Rys. 3.45. Deformacja oraz przemieszczenia węzłów w kierunku Z, podział belki na: jeden (a), dwa (b) oraz dziesięć (c) elementów skończonych



Rys. 3.46. Moment zginający M_y [kNm], podział belki na: jeden (a), dwa (b) oraz dziesięć (c) elementów skończonych



Rys. 3.47. Siły tnące V_z [kN], podział belki na: jeden (a), dwa (b) oraz dziesięć (c) elementów skończonych



Rys. 3.48. Ekstremalne wyężenia wg hipotezy HMH [MPa], podział belki na: jeden (a), dwa (b) oraz dziesięć (c) elementów skończonych

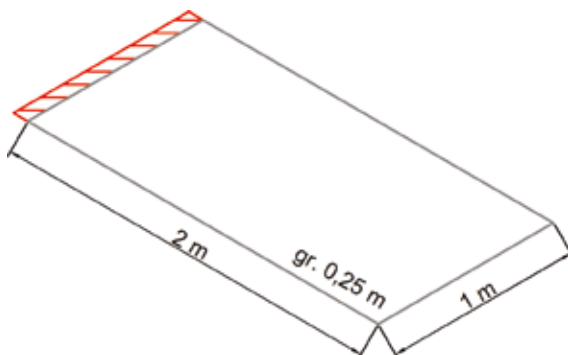
Praktyczne sprawdzenie poprawności konkretnej dyskretyzacji w obliczeniach MES najczęściej sprowadza się do typowego postępowania, jakim jest tzw. analiza zbieżności rozwiązania, rozumiana jako funkcja zagęszczania podziału układu na elementy skończone. Taka analiza przy ustalonym elemencie skończonym nazywa się badaniem zbieżności (udoskonaleniem) typu h . Polega ona na kontroli wyników rozwiązań w ustalonym punkcie danego układu w sekwencji kolejnych obliczeń z użyciem tego samego elementu, przy zagęszczaniu siatki podziału według ustalonej reguły, np. zwiększaniu dwukrotnym odpowiadającym połówkowaniu wymiaru elementu. Najczęściej w efekcie takich badań tworzona jest krzywa zbieżności wskazująca na wpływ np. ogólnej liczby stopni swobody czy liczby elementów zdyskretyzowanego układu na śledzony wynik rozwiązania, np. przemieszczenie wybranego punktu konstrukcji. Należy dodać, że w przypadku analizy nieliniowej zbieżność rozwiązania jest gorsza niż w badaniach odpowiedniego zagadnienia liniowego. Więcej na ten temat znaleźć można w pozycji CHRÓSCIELEWSKI I IN. [2004].

4. UKŁADY POWIERZCHNIOWE

Konstrukcje powierzchniowe w inżynierii, w tym – w budownictwie, są wykorzystywane równie często jak struktury prętowe. W kolejnych prostych przykładach zostaną omówione sposób definiowania danych, obliczeń i interpretacja wyników dla tego typu układów. Do tworzenia wszystkich składników tworzących model obliczeniowy, czyli materiałów, geometrii oraz obciążeń, zostanie wykorzystany moduł *Teddy* 🧩.

4.1. Materiały i definiowanie geometrii – SOFIMSHA

Przykład 4.1. Zbudować model obliczeniowy wspornika płytowego o długości $L = 2$ m, szerokości $S = 1$ m i grubości $G = 0,25$ m wykonanego z betonu C 45/55 (rys. 4.1).



Rys. 4.1. Wspornik płytowy – schemat statyczny

Również tym razem model obliczeniowy będzie tworzony z poziomu modułu *Teddy* 🧩. Tak jak w rozdziale 3, należy w oknie dialogowym modułu *Teddy* 🧩 przepisać omawiany dalej kod w języku problemowo zorientowanym programu SOFiSTiK. Sprawdzenia oraz obliczeń każdego utworzonego segmentu tego kodu, zapoczątkowanego komendą *+prog* i zakończonego komendą *end*, dokonuje się z poziomu modułu *WPS* 🖥️.

Definicja materiału przebiega w sposobu analogiczny do omówionego w rozdziale 3.1. Do tego celu wykorzystujemy moduł *AQUA*.

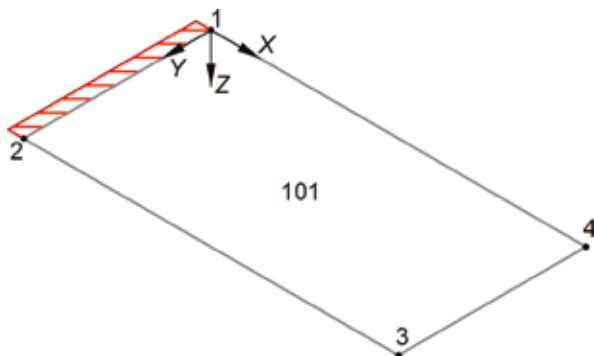
```
+prog aqua
head przekroje+materialy
echo full
$ norma
norm en 1993-2005 00 unit 5
$ material
conc 1 c 45 titl c45/55 $ beton
end
```

Do dyskretyzacji zagadnień powierzchniowych zostanie wykorzystany dwuwymiarowy, płaski, czterowęzłowy, powłokowy element skończony, który – podobnie jak omówiony wcześniej element prętowy – ma po sześć stopni swobody w każdym węźle (trzy translacje/ przesunięcia: $u-X$, $u-Y$, $u-Z$ oraz trzy parametry rotacyjne/ obrotowe: $\phi-X$, $\phi-Y$, $\phi-Z$). W przypadku zadań o charakterze płytowym problem można również rozwiązać, stosując elementy płytowe, które mają tylko po trzy stopnie swobody w każdym węźle (jedną translację/ ugięcie: $u-Z$ oraz dwa parametry rotacyjne/ obrotowe: $\phi-X$, $\phi-Y$).

Należy mieć świadomość, że w zagadnieniach powierzchniowych modelowanych z zastosowaniem dwuwymiarowych elementów skończonych poprawna dyskretyzacja układu jest trudniejsza oraz bardziej złożona i wymagająca niż w przypadku omówionych zagadnień prętowych opisywanych elementami jednowymiarowymi.

Przykład 4.1.1. Wariant traktujący wspornik jako pojedynczy element skończony

W celu ułatwienia rozpoczęcia pracy z programem – podkreślmy to stanowczo: tylko na potrzeby ilustracyjne postępowania – jako pierwszy wariant rozpatrzmy dyskretyzację z użyciem pojedynczego elementu skończonego (wręcz niedozwoloną w obliczeniach praktycznych) (rys. 4.2). Do stworzenia geometrii wspornika zostanie wykorzystany program *SOFIMSHA*. Pomoc/ instrukcję do tego modułu w zakresie generowania węzłów omówiono już w rozdziale 3.2.



Rys. 4.2. Wspornik płytowy – dyskretyzacja do celów ilustracyjnych z zastosowaniem tylko jednego elementu skończonego (niedozwoloną w obliczeniach praktycznych)

```
+prog sofimsha
head geometria
syst 3d $ 2dsl
```

Wywołanie programu *SOFIMSHA*.

Wybór przestrzeni zanurzenia układu dyskretyzowanego. Parametr *3d* definiuje układ jako strukturę przestrzenną (trójwymiarową w sensie intuicyjnym). Parametr *3d* wskazuje na potrzebę wykorzystania elementów prętowych, powłokowych lub bryłowych o maksymalnej dopuszczalnej przez system *SOFiSTiK* liczbie stopni swobody w węźle. Chcąc ograniczyć przykład do zagadnienia płytowego (płaskiego w sensie intuicyjnym), należałoby wybrać *2dsl* (rys. 3.10).

```

echo full
$ wezly definicja
node 1 0 0 0
      2 0 1 0
      3 2 1 0
      4 2 0 0
$ element definicja
quad no 101 n1 1 n2 2 n3 3 n4 4 mno 1 posi cent t 250

```

Definicja węzłów od nr 1 do 4, wg rys. 4.2.

Elementy powierzchniowe definiuje się komendą *quad*. Funkcja *no* określa nr elementu, tu 101. Funkcje *n1*, *n2*, *n3* i *n4* określają węzły elementu. Funkcją *mno* określa się materiał, tu nr 1 (beton C 45/55). Funkcją *posi* definiuje mimośród (zob. rys. 4.3), który w tym zadaniu nie jest wykorzystywany. Parametr *cent* oznacza, że powierzchnia środkowa elementu będzie się pokrywała z płaszczyzną węzłów. Funkcja *t* określa grubość, którą przy przyjętych ustawieniach (*unit 5* w komendzie *norm*) podaje się w mm.

Item	Description	Unit	Default
NO	Element number or mesh option	-	!
N1	Number of node 1	-	!
N2	Number of node 2	-	!
N3	Number of node 3	-	!
N4	Number of node 4	-	!
MNO	Element material number	-	*
MRF	Material number for reinforcements	-	*
NRA	Element Stiffness formulation	-	7
POSI	Element position w.r.t. the nodes	LIT	CENT
	CENT Element centred at nodes		
	ABOV Element above the nodes		
	BELO Element below the nodes		
T	Thickness t	[mm] ₁₀₁₀	*
T1	Thickness at node K1	[mm] ₁₀₁₀	T
T2	Thickness at node K2	[mm] ₁₀₁₀	T
T3	Thickness at node K3	[mm] ₁₀₁₀	T
T4	Thickness at node K4 (not for orthotropic plates)	[mm] ₁₀₁₀	T

QUAD

Numer elementu lub opcja *mesh*

Numery węzłów elementu
lub punkty strukturalne obszaru

Numer materiału

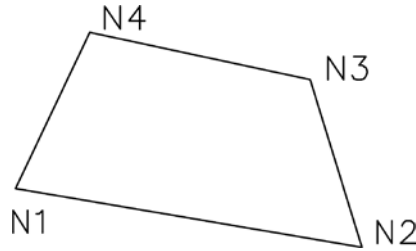
Mimośród:
cent – powierzchnia środkowa
w płaszczyźnie węzłów
abov – powierzchnia środkowa
przesunięta w górę o połowę
grubości
belo – powierzchnia środkowa
przesunięta w dół o połowę
grubości

Grubość elementu. Możliwość
stosowania elementów
o zmiennej grubości *T1–T4*

Rys. 4.3. Początek pomocy/ instrukcji do programu *SOFIMSHA* – funkcje komendy *quad*

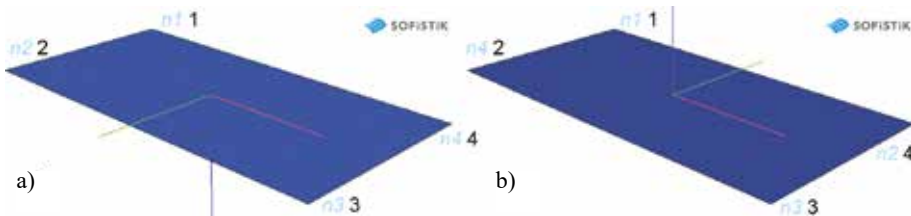
UWAGA. Podczas definiowania elementu powierzchniowego bardzo ważne jest przestrzeganie podawania kolejności węzłów, tutaj po funkcjach *n1*, *n2*, *n3* i *n4* (rys. 4.4).

Kolejność węzłów ustala lokalny układ współrzędnych elementu skończonego. Ogólnie mówiąc, w kontekście elementu wzorcowego/ macierzystego (układ *rst*), w programie SOFiSTiK przyjęto orientację układu taką, że (rys. 4.5): lokalna oś *x* (*r*) powiązana jest z bokiem określonym węzłami *n4* i *n3*, oś *y* (*t*) – z bokiem określonym węzłami *n3* i *n2*, natomiast oś *z* (*t*) jest do nich ortogonalna z zachowaniem prawoskrętności układu zaczepionego w środku geometrycznym elementu.



Rys. 4.4. Pomoc/ instrukcja do programu *SOFIMSHA* – numeracja węzłów czterowęzłowego elementu powierzchniowego, orientacja lokalnej osi z do rysunku (w dół)

Należy zwrócić uwagę, że sposób zorientowania lokalnego układu współrzędnych rst przyjęty w SOFiSTiK jest inny niż w klasycznych elementach izoparametrycznych (zob. np. BATHE [1996]). W uzupełnieniu należy dodać, że obraz/ odwzorowanie układu osi rst z elementu macierzystego/ wzorcowego w przestrzeni fizycznej zanurzenia elementu w ogólnym przypadku jest układem krzywoliniowym (tu skośnym), a więc obraz rst nie jest tożsamy z lokalnym ortonormalnym układem osi xyz elementu. Reasumując, kolejność podawania węzłów po funkcjach $n1$, $n2$, $n3$ i $n4$ musi być obiegowa po obwodzie, a kierunek obiegu, przeciwny do ruchu wskazówek zegara, pokazuje orientację i w programie SOFiSTiK ujemny zwrot lokalnej osi z elementu skończonego w przestrzeni fizycznej. Należy mieć to na względzie, ponieważ lokalny układ współrzędnych xyz jest wykorzystywany do definicji sił wewnętrznych powłoki. Warto zatem przestrzegać zasady, że w całym układzie, o ile to możliwe, stosujemy jeden ustalony kierunek lokalnej osi z , najlepiej zgodny z osią globalną Z .



Rys. 4.5. Schemat układu – numeracja węzłów podana przeciwnie do kierunku wskazówek zegara (a), numeracja węzłów podana zgodnie z kierunkiem wskazówek zegara (b)


W rozważanym przykładzie, przy zastosowaniu kierunku przeciwnego do ruchu wskazówek zegara ($n1$ 1 $n2$ 2 $n3$ 3 $n4$ 4), oś lokalna z elementu skierowana jest w dół (rys. 4.5a), a przy podawaniu węzłów zgodnie z kierunkiem ruchu wskazówek zegara ($n1$ 1 $n2$ 4 $n3$ 3 $n4$ 2), oś lokalna z elementu skierowana jest w górę (rys. 4.5b). Wybór orientacji układu lokalnego nie powoduje błędów obliczeniowych, może być jednak kłopotliwy dla użytkownika przy generacji obciążenia lub interpretacji wyników. Przy zastosowaniu nazewnictwa klasycznej teorii płyty/ belek zwrot lokalnej osi z określa spód płyty/ elementu (belki) – przy z skierowanej w górę element jest odwrócony (spód u góry, wierzch u dołu).

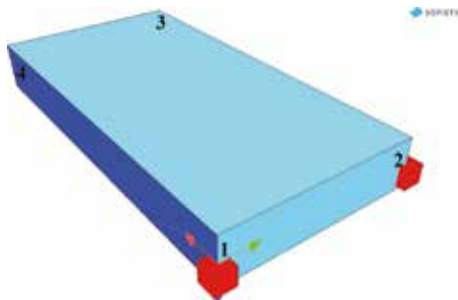
\$ podpory

```
node 1 - - - fix f
```

```
      2 - - - fix f
end
```

Definicja warunków brzegowych (podparcia). Pełne utwierdzenie przypisujemy węzłom nr 1 i 2. W klasycznej MES więzy przypisuje się tylko do węzłów, a nie do krawędzi elementów.

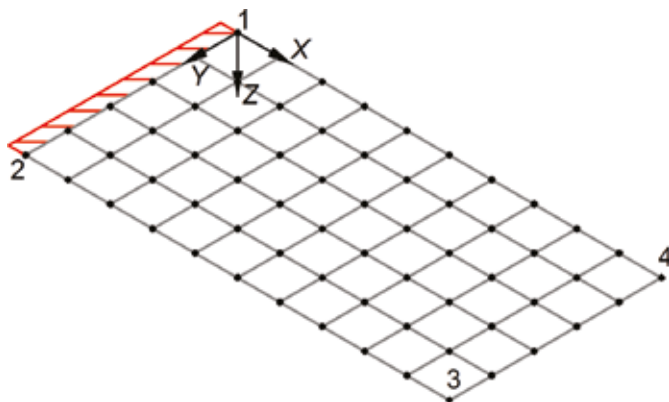
Wizualizację wspornika opisanego jednym elementem skończonym (nie dozwoloną w obliczeniach praktycznych), stworzoną w module *Animator* , przedstawiono na rys. 4.6.



Rys. 4.6. Moduł *Animator* – wizualizacja wspornika opisanego jednym elementem skończonym (nie dozwoloną w obliczeniach praktycznych)

Przykład 4.1.2. Wariant wspornika o typowym podziale na elementy

Utworzyć model obliczeniowy wspornika płytowego (rys. 4.1), stosując podział na 50 elementów (10×5) rozpięty na siatce 66 węzłów (11×6) (rys. 4.7).



Rys. 4.7. Wspornik płytowy – dyskretyzacja na 10×5 elementów czterowęzłowych, oznaczenie naroży układu, tzw. punktów/ węzłów strukturalnych

Geometria układu ponownie zostanie utworzona w module *SOFIMSHA*. Do stworzenia regularnej tzw. siatki strukturalnej podziału elementów skończonych (tu 10×5) zostanie użyty wewnętrzny generator komendy *quad* (opcja *mesh*), który do definicji dyskretyzowanego

obszaru wykorzystuje narożnikowe tzw. punkty/ węzły strukturalne (tu obszaru topologicznie równoważnego prostokąta, tzw. prymitywu, zob. np. CHRÓŚCIELEWSKI I IN. [2014]). Stąd początek kodu jest identyczny jak w przykładzie 4.1.1, ponieważ zdefiniowane tam węzły (od nr. 1 do nr. 4) mają charakter punktów strukturalnych, zatem zmiany wystarczy wprowadzić w miejscu definicji elementów i dalej warunków brzegowych.

Opcja *mesh* komendy *quad* pozwala w prosty sposób stworzyć siatkę obszaru ograniczoną punktami/ węzłami strukturalnymi $n1$, $n2$, $n3$ i $n4$. Za pomocą dwóch funkcji m oraz n określa się gęstość podziału przeciwległych boków na równe odcinki (rys. 4.8). Następnie punkty na tych bokach łączy się liniami, tworząc regularną tzw. siatkę strukturalną, gdzie jednoznacznie określona jest struktura siatki podziału wynikająca z odwzorowania jej z topologii, tzw. prymitywu. Ten sposób jest bardzo wygodny, ale możliwy do zastosowania tylko w obszarach regularnych, dla których opracowane są prymitywy. Niestety traci się bezpośrednio możliwość kontroli numeracji węzłów i elementów. Dlatego też przy definicji utwierdzenia krawędzi wspornika niewygodnie byłoby przypisywanie warunków brzegowych do konkretnych węzłów. Lepiej zastosować komendę *boun*, za pomocą której można przypisać więzy do węzłów tworzących szereg opisany tu punktami strukturalnymi.

M	Subdivison of edge N1-N2,N3-N4	-	1/0
N	Subdivison of edge N2-N3,N4-N1	-	1/0
M12	optional middle node of edge N1-N2	-	-
M23	optional middle node of edge N2-N3	-	-
M34	optional middle node of edge N3-N4	-	-
M41	optional middle node of edge N4-N1	-	-

Podziały krawędzi

Rys. 4.8. Ciąg dalszy pomocy/ instrukcji do programu *SOFIMSHA*
– wybrane funkcje komendy *quad*

\$ elementy definicja

```
quad mesh n1 1 n2 2 n3 3 n4 4 mno 1 posi cent t 250 m 5 n 10
```

\$ podpory

```
boun 101 titl utwierdzenie
```

Stosując podporę liniową, w pierwszej kolejności nadaje się jej numer i nazwę.

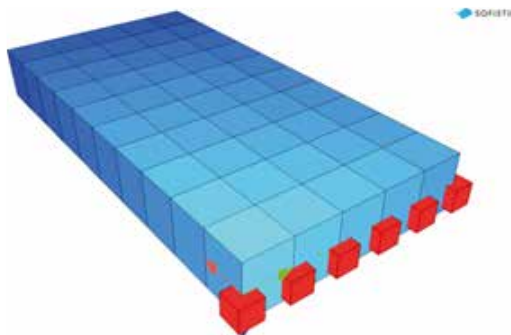
```
boun na 1 ne 2 nm fit type f
```

Komenda *boun* przypisuje nałożenie warunków brzegowych (podparcia) węzłom znajdującym się w szeregu pomiędzy punktem strukturalnym początkowym *na* a punktem końcowym *ne*. Opcja *fit* przypisuje warunki brzegowe wszystkim węzłom w szeregu z pominięciem algorytmu liniowej periodyczności określanej przyrostem numeru węzła (rys. 4.9). Wartość funkcji *type* przyjęto jako *f* (dla podpór stałych obowiązują typy jak w komendzie *node*, rys. 3.13).

BOUN			
Item	Description	Unit	Default
NA	Starting node or element number	-	!
NE	End node number	-	
NM	Increment of numbers or Literal FIT / FITL	-/LIT	
TYPE	Direction of support	LIT	
	CX in global X-direction		
	CY in global Y-direction		
	CZ in global Z-direction		
	CN in longitudinal direction		
	CT in both transverse directions		
	DX Rotation about global X		
	DY Rotation about global Y		
	DZ Rotation about global Z		
	DN Rotation about longitudinal		
	DT Rotation about both transverse di- rections		
CA	Bedding constants at the beginning	[*].	
CE	Bedding constants at the end	[*].	CA

Rys. 4.9. Pomoc/ instrukcja do programu SOFIMSHA – wybrane funkcje komendy *boun*

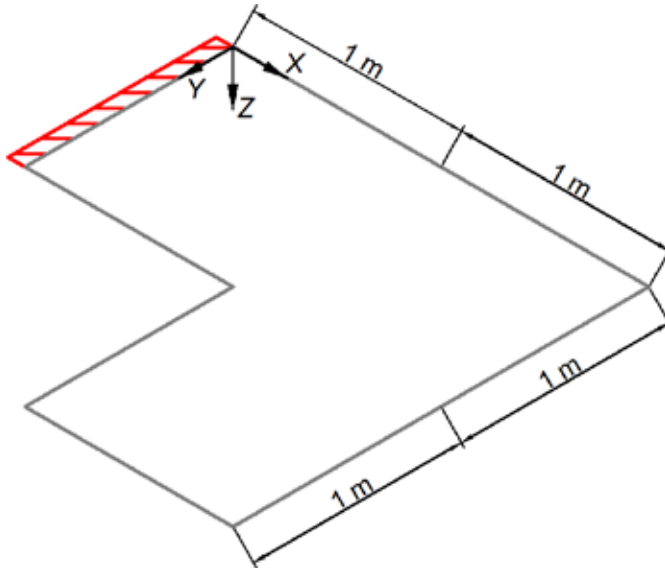
Wizualizację wygenerowanego modelu obliczeniowego wspornika płytowego o zagęszczonej siatce podziału na elementy w module *Animator* przedstawiono na rys. 4.10.

Rys. 4.10. Moduł *Animator* – wizualizacja modelu obliczeniowego wspornika płytowego z dyskretyzacją na 50 elementów skończonych

Zadanie 2. Modyfikacja geometrii belki do wspornika płytowego typu L

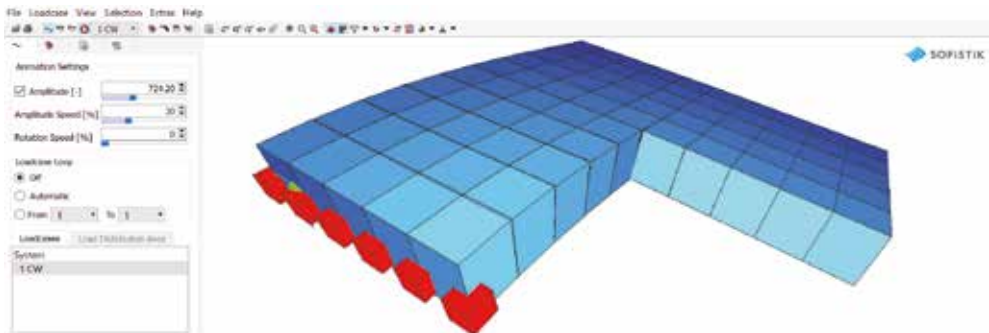
W ramach treningu rozbudować wprowadzoną w zadaniu 4.1.2 geometrię prostego wspornika do wspornika w kształcie litery L wg rys. 4.11. Wykorzystać koncepcję podziału układu na podobzary równoległoboków dających się odwzorować topologicznie w prostokąt jako prymityw. Stosując opcję *mesh* w komendzie *quad*, należy pamiętać o tym, by podziały *m* i *n* ustalić tak, aby długości boków elementów na krawędziach wspólnych podobzarów

prostokątnych (odpowiadającym prymitywom) były jednakowe, to znaczy tak, aby zachowana była spójna siatka podziału na elementy skończone i nie pojawiły się tzw. węzły sieroce. Następnie przeliczyć układ obciążony ciężarem własnym.



Rys. 4.11. Wspornik płytowy typu L – schemat statyczny, obciążenie ciężarem własnym

Wizualizację w konfiguracji zdeformowanej utworzoną w module *Animator* przedstawiono na rys. 4.12.



Rys. 4.12. Moduł *Animator* – wspornik płytowy typu L obciążony ciężarem własnym, wizualizacja



4.2. Obciążenia, obliczenia statyczne

Wykorzystując dyskretyzację wspornika płytowego o zagęszczonym podziale z zadania 4.1.2 (rozdział 4.1), wygenerować następujące obciążenia działające na:

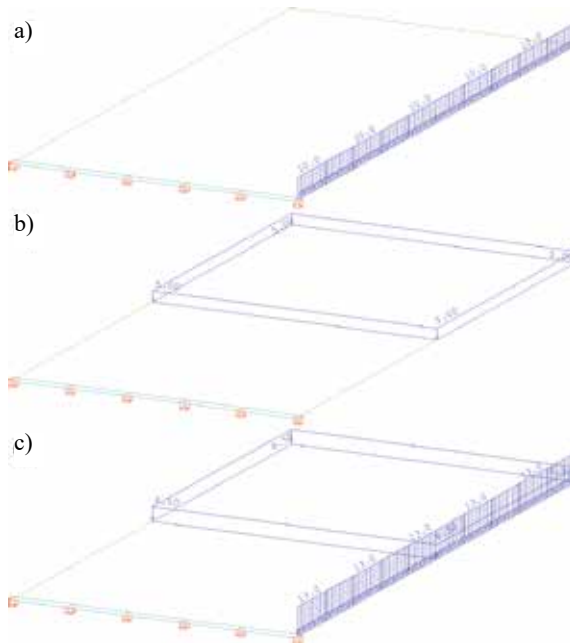
- a) obciążenie nr 1, liniowe na krawędzi bocznej wspornika o wartości 10 kN/m;
 - b) obciążenie nr 2, powierzchniowe na ½ obszaru wspornika o wartości 5 kN/m;
 - c) obciążenie nr 3, sumaryczne z przypadków nr 1 i nr 2 z mnożnikiem 1,3.
- Następnie dla stworzonych obciążeń wykonać obliczenia statyczne.

W przypadku konstrukcji powierzchniowych obowiązują identyczne zasady jak w układach prętowych. Ponownie można wykorzystać moduł *SOFILOAD*, stosując opis obciążenia typu strukturalnego i/lub wolnego. W tym zadaniu obciążenia generuje się jako typu wolnego powiązane z przestrzenią, a nie konstrukcją.

<pre>+prog sofiload head obciazenia lc 1 titl obc_A_linowe line qgrp type pzz p1 10 x1 0 y1 1 x2 2 y2 1 lc 2 titl obc_B_powierzchniowe area qgrp type pzz p1 5 x1 1 y1 0 x2 2 y2 0 x3 2 y3 1 x4 1 y4 1 lc 3 titl obc_C_suma copy 1,2 fact 1.3 end</pre>	<p>Wywołanie programu <i>SOFILOAD</i>.</p> <p>Stworzenie przypadku obciążenia nr 1.</p> <p>Definiowanie obciążenia liniowego jest analogiczne jak w układach prętowych. Jedyna różnica występuje w opcji <i>qgrp</i> funkcji <i>ref</i> –wskazującej na przeszukiwanie powiązań z konkretnymi elementami powłokowymi.</p> <p>Stworzenie przypadku obciążenia nr 2.</p> <p>Obciążenia powierzchniowe zadaje się komendą <i>area</i>. Definiowanie wygląda podobnie jak w przypadku komendy <i>line</i>. Różnica leży w konieczności podania większej liczby punktów ograniczających obszar działania obciążenia.</p> <p>Stworzenie przypadku obciążenia nr 3, sumy obciążeń nr 1 i nr 2 z mnożnikiem 1,3.</p> <p>Kombinacja liniowa/ superpozycja obciążeń nr 1 i nr 2 jest tworzona z wykorzystaniem komendy <i>copy</i>.</p> <p>Zamknięcie programu <i>SOFILOAD</i>.</p>
--	---

Zakończenie zadania wymaga uruchomienia obliczeń modułów *SOFILOAD* i *ASE*, co wykonuje się z poziomu programu *WPS* . Poszczególne przypadki obciążeń zobrazowano z wykorzystaniem modułu *WinGraf*  na rys. 4.13.

<pre>+prog ase head obc_obliczenia lc (1 3 1) end</pre>	<p>Wywołanie programu <i>ASE</i>.</p> <p>Obliczenia statyczne przypadków obciążeń od nr. 1 do nr. 3 ze zmianą kroku o 1.</p> <p>Zamknięcie programu <i>ASE</i>.</p>
--	---



Rys. 4.13. Obciążenia działające na wspornik płytowy – przypadki nr 1 (a), nr 2 (b) oraz nr 3 (c)

4.3. Prezentacja wyników

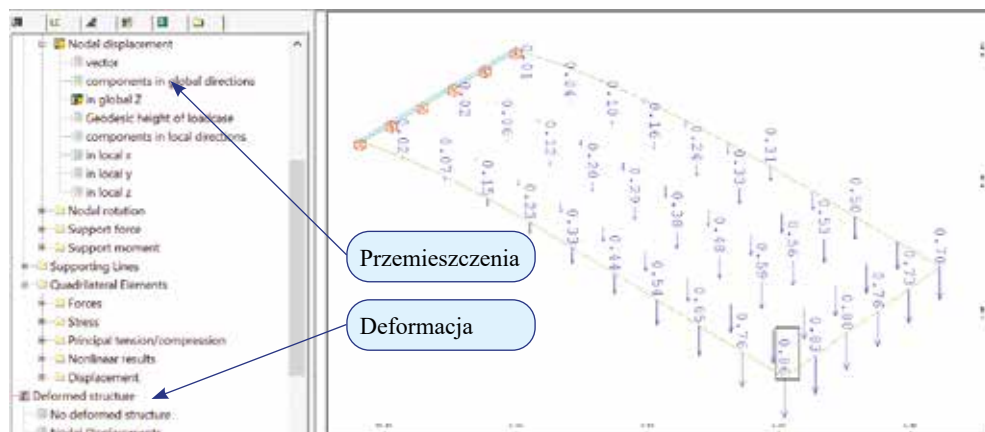
W układach powierzchniowych do prezentacji wyników można wykorzystać te same moduły, które stosowano w przypadku ustrojów prętowych i opisano w rozdziale 3.4. W bieżącym zadaniu zostanie wykorzystany głównie moduł *WinGraf*.

Wybrane wyniki rozwiązania przykładu 4.1.2 od przypadku obciążenia nr 3 pokazano na kolejnych rysunkach: ugięcia płyty (pionowe przemieszczenia węzłów) na zdeformowanej konfiguracji układu (rys. 4.14), pole m_{xx} składowej momentów o wektorze równoległym do lokalnej osi y , która wywołuje zginanie wzdłuż lokalnej osi x (rys. 4.15), oraz pole składowej normalnej naprężeń σ_{xx} w kierunku lokalnej osi x na spodzie płyty (rys. 4.16).

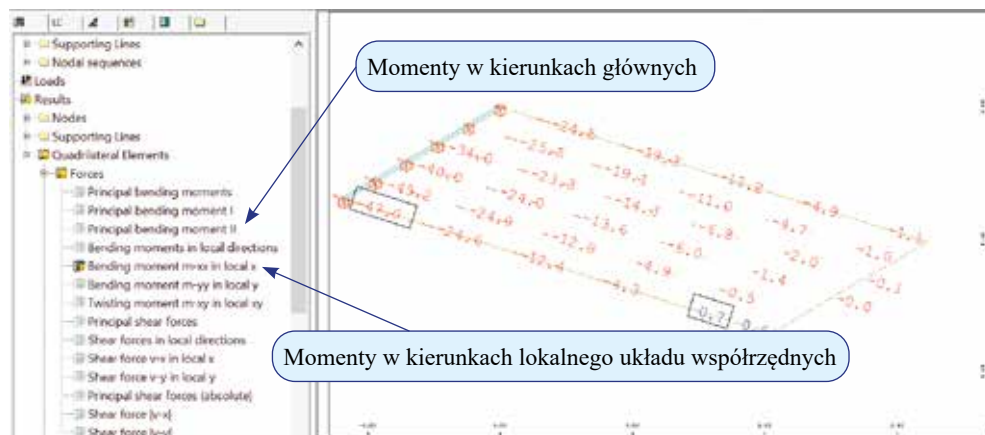
Należy zauważyć, że w przykładzie 4.1.2 osie lokalne xyz wszystkich elementów są kolinearne z globalnym układem współrzędnych XYZ . Ponadto osie x są równoległe do dłuższej krawędzi wspornika (rys. 4.17). Taki sposób przyjmowania układów lokalnych xyz bardzo ułatwia interpretację wszystkich wielkości, które są zdefiniowane względem tych układów.

UWAGA. Poprawna interpretacja wyników zawsze, a szczególnie w przypadku elementów powłokowych, wymaga: po pierwsze, znajomości orientacji lokalnych układów współrzędnych xyz względem układu osi globalnego XYZ ; po drugie, znajomości definicji przekrojowych sił wewnętrznych w elemencie (tu powłokowym, rys. 4.18). Należy zwrócić uwagę, że kierunki i zwroty wektorów składowych momentów zginających (m_{xx} , m_{yy} , m_{xy} , rys. 4.18a) są inne niż znakowanie sił membranowych (n_{xx} , n_{yy} , n_{xy} , rys. 4.18b), które tak jak naprężenia (σ_{xx} , σ_{yy} , $\sigma_{xy} \equiv \tau_{xy}$) są zgodne z układem osi lokalnych xy . Konwencja oznaczeń składowych momentów jest konsekwencją zachowania zgodności składowych naprężenia w powłokach

przy bezpośrednim superponowaniu stanu zginania ze stanem membranowym. Stąd, zgodnie z założeniem klasycznej teorii powłok (zob. np. RADWAŃSKA, M. [2013]), przyjmuje się za dodatnie te składowe, które powodują rozciąganie spodu płyty; wówczas drugi indeks x lub y składowych m_{xx} i m_{yy} wskazuje powodowany przez nie kierunek zginania walcowego (zob. rys. 4.18).

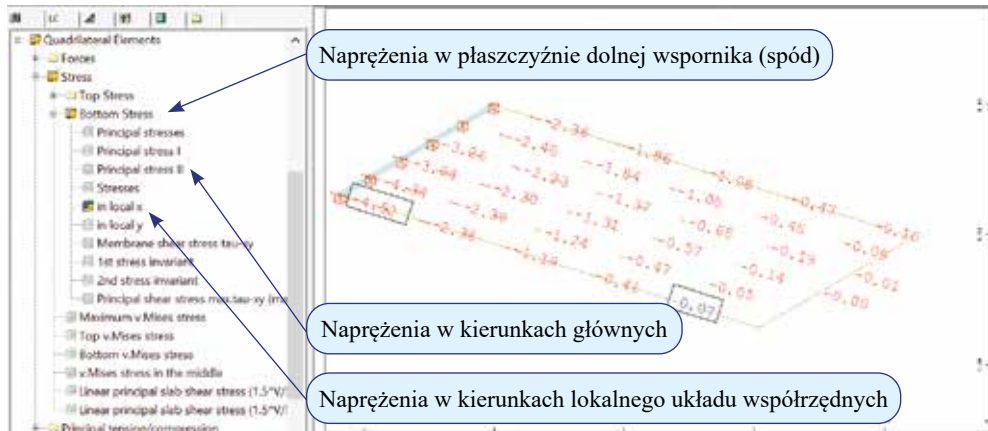


Rys. 4.14. Moduł *WinGraf* – deformacja układu, ugięcia w węzłach [mm], obciążenie nr 3

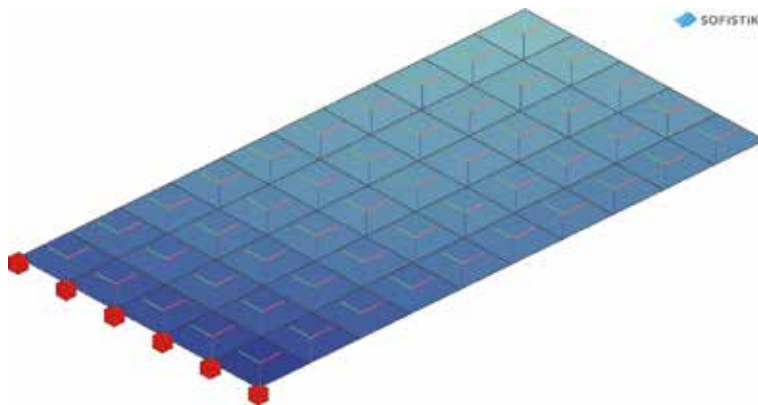


Rys. 4.15. Moduł *WinGraf* – składowa m_{xx} momentu o wektorze równoległym do osi y , która wywołuje zginanie wzdłuż lokalnej osi x [kNm/m], obciążenie nr 3

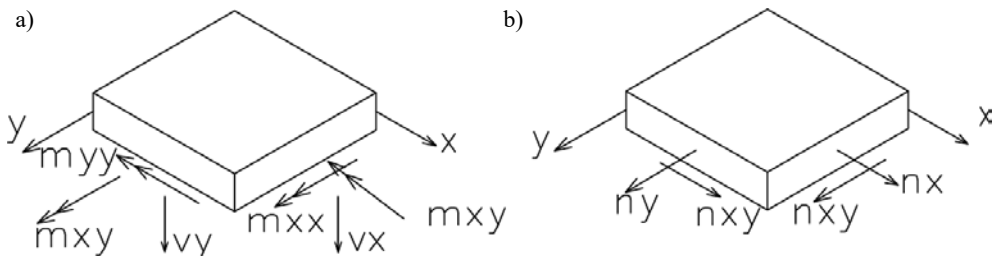
W przypadku elementów powierzchniowych warto poświęcić więcej uwagi sposobowi prezentacji wyników. Na rys. 4.15 oraz rys. 4.16 wykorzystano domyślne ustawienia modułu *WinGraf* – zaprezentowano składowe wektorów dla wartości węzłowych. Nie jest to jednak jedyny sposób, w programie oferowane są także inne możliwości. Chcąc zmienić ustawienia, należy wykorzystać trzecią zakładkę drzewa wyboru – reprezentacja. Na rys. 4.19 przedstawiono jeszcze raz naprężenia normalne na kierunku lokalnej osi x spodu wspornika, tym razem jako mapę (izopowierzchnię) wygenerowaną z wartości węzłowych.



Rys. 4.16. Moduł *WinGraf* – składowa normalna napężenia σ_{xx} w kierunku lokalnej osi x , na spodzie płyty [MPa], obciążenie nr 3



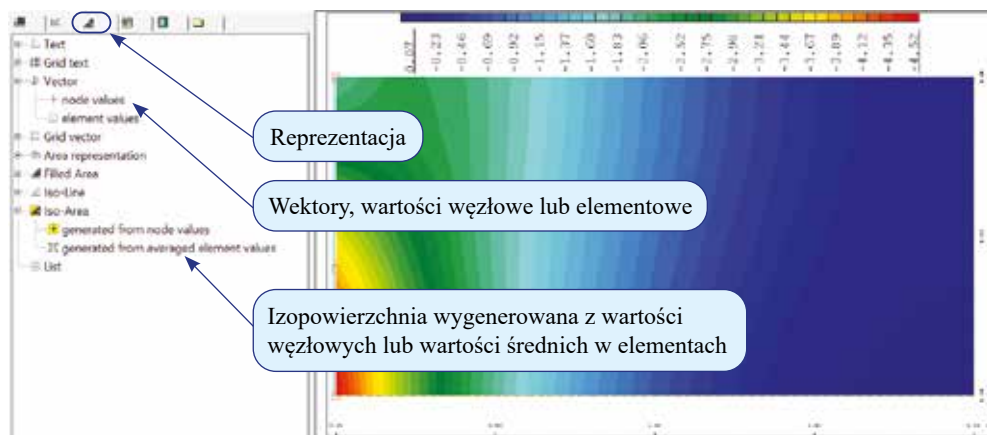
Rys. 4.17. Moduł *Animator* – lokalne układy współrzędnych: x – czerwone, y – zielone i z – granatowe



Rys. 4.18. Pomoc/ instrukcja do programu *SOFIMSHA* – konwencja znakowania sił wewnętrznych w elementach powłokowych, momentów i sił poprzecznych (a) oraz sił membranowych (b)

UWAGA. W opcjach wyboru prezentacji wyników pojawiają się wielkości węzłowe i średnio-elementowe. Należy zdawać sobie sprawę z tego, że w programach MES wartości sił we-

wewnętrznych i naprężenia najczęściej nie są obliczane w żadnych z tych miejsc. Wartości te oblicza się na ogół w tzw. punktach całkowania wielkości elementowych (przy tzw. pełnym całkowaniu (FI) elementów czterowęzłowych klasy C^0 są to cztery punkty w obszarze pojedynczego elementu). Węzły elementu i punkty całkowania nie pokrywają się, chociaż w pewnych przypadkach mogą być tożsame. W programie SOFISTIK, aby uprościć to zagadnienie, obliczone wartości są uśredniane z odpowiednimi wagami do węzłów lub elementów. Przy dobrze zdefiniowanej siatce elementów skończonych różnica między tymi dwiema wersjami nie powinna być znacząca.



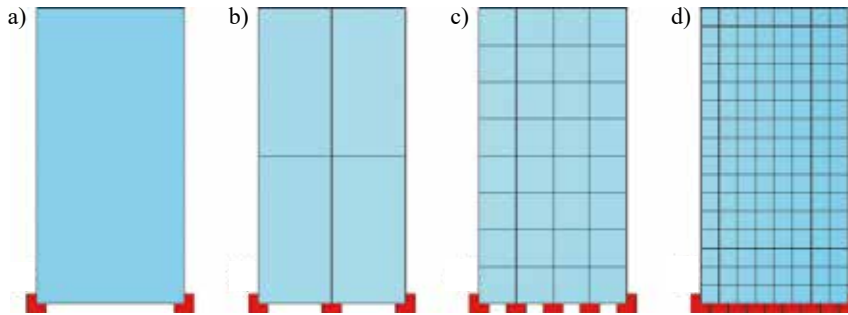
Rys. 4.19. Moduł *WinGraf* – mapa składowej normalnej σ_{xx} naprężeń w kierunku lokalnej osi x , na spodzie płyty [MPa], obciążenie nr 3

4.4. Uwagi o zbieżności rozwiązań MES, udoskonalenie typu h

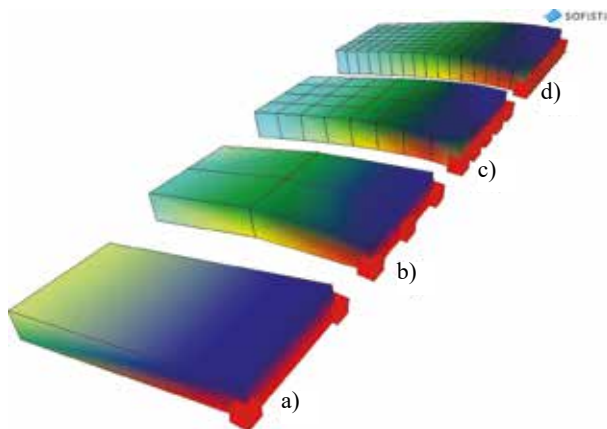
W dotychczasowych rozważaniach dotyczących stosowania elementów powierzchniowych skupiono się na umiejętności wygenerowania tzw. modelu obliczeniowego bez analizy wpływu na rozwiązanie gęstości siatki podziału układu na elementy skończone. Jak podkreślono w rozdziale 3.8, nie wolno nigdy zapominać, że **metoda elementów skończonych jest metodą przybliżoną!** Spełnienie kryteriów zbieżności w przypadku powierzchniowych elementów skończonych jest zaś zagadnieniem bardziej wymagającym niż w przypadku elementów prętowych. Zagadnienie to, na przykładzie badania zbieżności typu h poprzez zagęszczanie podziału na ustalone elementy skończone, zasygnalizujemy w dalszej części tego rozdziału.

W przykładach wykorzystano stosowane w przykładzie 4.1 dwuwymiarowe, czterowęzłowe powłokowe elementy skończone typu Timoszenko-Reissnera o sześciu stopniach swobody w węźle (trzy translacje/ przesunięcia: $u-X$, $u-Y$, $u-Z$ oraz trzy parametry rotacyjne/obrotowe: $\phi-X$, $\phi-Y$, $\phi-Z$). Są to elementy klasy C^0 o dwuliniowych funkcjach kształtu ze wzbogaceniem stanu deformacji w powierzchni, eliminującym efekt blokady. Uwzględniają one efekt ścinania oraz zróżnicowanie położenia powierzchni odniesienia powłoki (mi-mośród).

Wpływ gęstości podziału na rozwiązanie analizowanego wspornika przeanalizujemy na podstawie czterech dyskretyzacji, stosując tzw. podział strukturalny (regularny) na: $1 \times 1 = 1$ (a), $2 \times 2 = 4$ (b), $8 \times 4 = 32$ (c) oraz $16 \times 8 = 128$ (d) elementów skończonych (rys. 4.20). Wszystkie inne parametry pozostawiono bez zmian. Model obciążono, wykorzystując przypadek nr 3 i obciążenie typu wolnego z rozdziału 4.2.

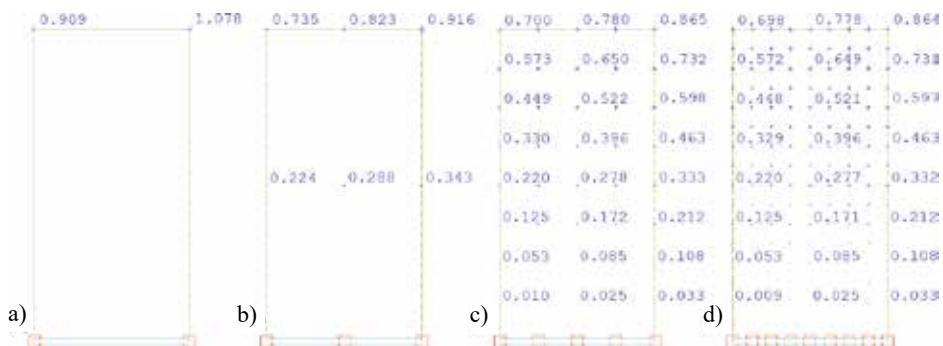


Rys. 4.20. Wspornik płytowy, podziały na: $1 \times 1 = 1$ (a), $2 \times 2 = 4$ (b), $8 \times 4 = 32$ (c), $16 \times 8 = 128$ (d) powłokowych czterowęzłowych elementów skończonych

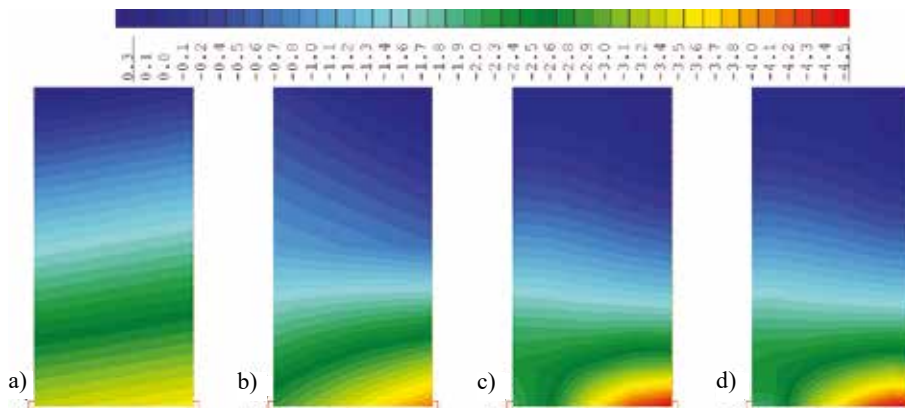


Rys. 4.21. Wizualizacja deformacji wspornika płytowego, podziały na: $1 \times 1 = 1$ (a), $2 \times 2 = 4$ (b), $8 \times 4 = 32$ (c), $16 \times 8 = 128$ (d) powłokowych czterowęzłowych elementów skończonych, obciążenie nr 3

Pierwszy wynik, jaki zaprezentowano, to wizualizacja deformacji wspornika (rys. 4.21). W rozwiązaniach (a) oraz (b) wyraźnie uwidacznia się stosowanie w elementach dwulinowych funkcji kształtu, dopiero w wynikach (c) i (d) daje się zaobserwować spodziewane rozwiązania (zakrzywienie krawędzi po długości wspornika). Na rys. 4.22 przedstawiono pionowe przemieszczenia węzłów, wartości maksymalne (w prawym górnym narożniku wspornika) stabilizują się dopiero w przypadku rozwiązań (c) oraz (d). Ostatni przedstawiony wynik to mapy naprężenia na kierunku lokalnej osi x spodu wspornika (rys. 4.23). W tym przypadku również, aby uzyskać poprawne rozwiązanie, niezbędny jest wyższy poziom dyskretyzacji. Powtarzalne wyniki otrzymuje się dopiero w przypadku rozwiązań (c) oraz (d).





Rys. 4.22. Przemieszczenia (ugięcie) węzłów w kierunku Z [mm] wspornika płytowego, podziały na:
 $1 \times 1 = 1$ (a), $2 \times 2 = 4$ (b), $8 \times 4 = 32$ (c), $16 \times 8 = 128$ (d)
 powłokowych czterowęzłowych elementów skończonych, obciążenie nr 3



Rys. 4.23. Naprężenia na kierunku lokalnej osi x spodu wspornika płytowego [MPa], podziały na:
 $1 \times 1 = 1$ (a), $2 \times 2 = 4$ (b), $8 \times 4 = 32$ (c), $16 \times 8 = 128$ (d)
 powłokowych czterowęzłowych elementów skończonych, obciążenie nr 3

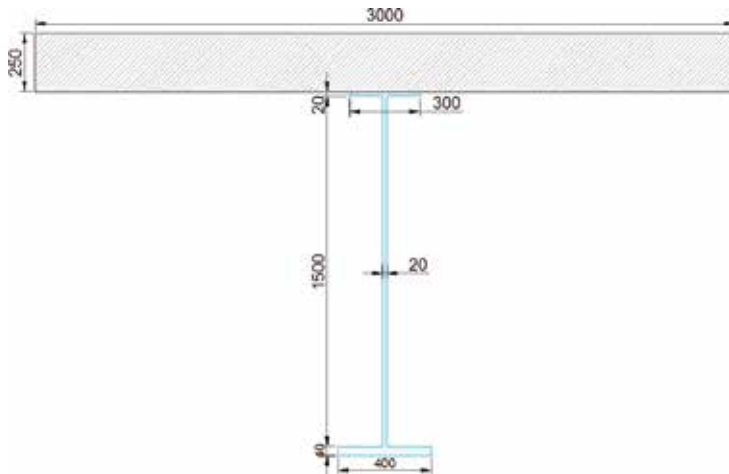
Odpowiedź na pytanie, jaki jest prawidłowy poziom dyskretyzacji modelu obliczeniowego, tak jak w przypadku elementów prętowych, daje analiza zbieżności podziału na elementy skończone. Na podstawie zaprezentowanych wyników można stwierdzić, że podział wspornika na 32 elementy skończone daje wystarczająco dokładne rozwiązanie, jednocześnie nie wymagając dużego nakładu czasowego potrzebnego do obliczeń.

5. UKŁADY POWIERZCHNIOWO-PRĘTOWE

Kolejnym etapem rozpoznawania możliwości systemu SOFiSTiK są układy, w których jednocześnie wykorzystuje się elementy powierzchniowe oraz prętowe. Dyskretyzacja układu będzie tworzona na trzy różne sposoby: z poziomu modułu *Teddy* , przy wykorzystaniu programu *SOFIMSHA* i *SOFIMSHC* oraz za pomocą modułu *Sofiphus-X* .



5.1. Materiały, przekroje, geometria – SOFIMSHA

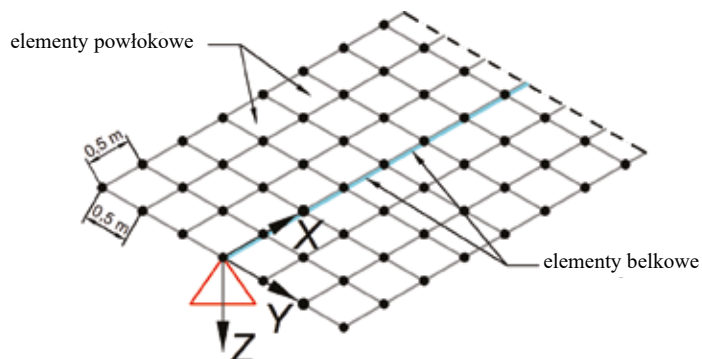
Przykład 5.1. Stworzyć model obliczeniowy belki swobodnie podpartej o długości $L = 20\text{ m}$ i przekroju zespolonym stalowo-betonowym (S 355, C 35/45) z rys. 5.1.



Rys. 5.1. Przekrój belki zespolonej

Do dyskretyzacji płyty betonowej użyć elementów powłokowych, dźwigar stalowy opisać elementami prętowymi/ belkowymi (rys. 5.2). Zastosować podparcie typowe dla belki swobodnie podpartej, pamiętając o dodatkowej blokadzie obrotu ϕ -X zapobiegającej ruchowi sztywnemu (obrót belki wokół własnej osi). Przeprowadzić obliczenia i przeanalizować wyniki powodowane ciężarem własnym. Dane przygotować w programie *SOFIMSHA*.

Przygotowanie kodu nowych danych prowadzone będzie z poziomu modułu *Teddy* , a do obliczeń na postawie tego kodu zostanie wykorzystany program *WPS* .

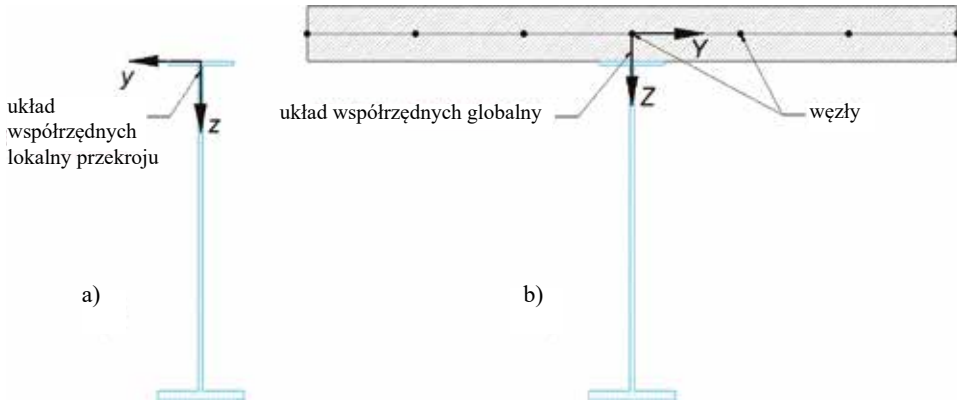


Rys. 5.2. Dyskretyzacja układu belki zespolonej

Pierwszym etapem rozwiązania zadania jest definicja materiału i przekrojów, w której zostanie użyty program *AQUA*, analogicznie do sposobu przedstawionego w rozdziale 3.1. Jako regułę zaleca się skopiowanie kodu danych utworzonego i sprawdzonego wcześniej do zadań o zbliżonych parametrach, a następnie zmodyfikowanie go do zakresu aktualnego wariantu. Takie podejście pozwala uniknąć szeregu drobnych błędów edycyjnych.

```
+prog aqua
head przekroje+materiały
echo full
norm en 1993-2005 00 unit 5 $ norma
$ material
ste 1 s 355 titl s355 $ stal
conc 2 c 35 titl c35/45 $ beton
```

UWAGA. Istotnym problemem rzutującym na późniejsze postępowanie jest odpowiednie, wymagające przemyślenia, przyjęcie globalnego i lokalnych układów współrzędnych poszczególnych grup elementów skończonych. Od lokalizacji tych układów zależą będą definiowanie sprzęgających je dystansów/ mimośrodków, usytuowanie warunków brzegowych oraz lokalizacja obciążenia. W elementach prętowych mimośrodek, a poprawniej: dystans, na podstawie którego oblicza się mimośrodek, jest odmierzany od początku układu współrzędnych (rys. 5.3a) wprowadzanego na etapie jego definiowania, a nie względem środka ciężkości przekroju, jak formalnie jest definiowany. W elementach powłokowych dopuszczalne są trzy położenia powierzchni odniesienia, tj. powierzchnia środkowa lub górna, lub dolna strona płyty. Położenie górne i dolne definiują dwa dopuszczalne mimośrodky mierzone względem powierzchni środkowej. W zadaniu przyjęto założenia pokazane na rys. 5.3. Przy tych założeniach w belkach stalowych należy uwzględnić dystans/ mimośrodek o wartości połowy grubości płyty, natomiast nie trzeba wprowadzać go w płycie, gdzie jako powierzchnię odniesienia przyjęto płaszczyznę środkową.



Rys. 5.3. Umieszczenie początku lokalnego układu współrzędnych (y,z) w przekroju stalowym (a) oraz układu (Y,Z) zgodnego z płaszczyzną siatki węzłów w powierzchni środkowej płyty (b)

§ przekroj poprzeczny

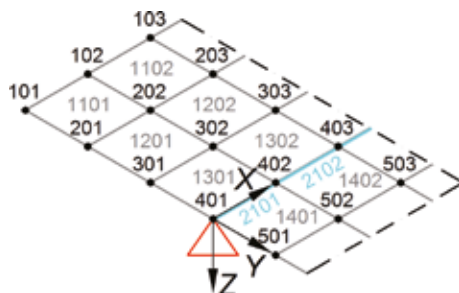
```
sect 1 mno 1 zm 250/2 titl przekroj_a
```

Definicja przekroju, uwzględniająca dystans/mimośród w kierunku lokalnej osi z (rys. 5.3a) o wartości $250/2 = 125$ mm.

```
poly type opz
vert 1 0 0
      2 150 0
      3 150 20
      4 10 20
      5 10 1520
      6 200 1520
      7 200 1560
      8 0 1560
```

```
end
```

W tym wariancie zadania do stworzenia geometrii belki zostanie wykorzystany program *SOFIMSHA*. W pierwszej kolejności zostanie wygenerowana siatka elementów płyty betonowej, a związany z nią układ współrzędnych będzie traktowany jako układ globalny całego ustroju. Następnie do odpowiednich węzłów elementów powierzchniowych zostaną dołączone belki z uwzględnieniem dystansu/ mimośrodu. Niestety wszystkie węzły oraz elementy muszą zostać wygenerowane samodzielnie, gdyż stosowanie opcji *mesh* w komendzie *quad* pozbawia użytkownika kontroli nad przebiegającą automatycznie numeracją. Kontrola numeracji węzłów w tym przykładzie jest konieczna i wynika z potrzeby samodzielnego połączenia dwóch typów elementów przez użytkownika. Najlepiej tak ją rozplanować, aby możliwe było jak najdalej posunięte wykorzystanie komendy pętli, szeregu i funkcji zmiennej. Przyjętą w tym przykładzie koncepcję numeracji węzłów i elementów przedstawiono na rys. 5.4; założono podział płyty na 40×6 (po długości \times po szerokości) elementów skończonych.



Rys. 5.4. Przyjęta koncepcja numeracji węzłów i elementów w kontekście użycia komendy pętli i szeregu

```
+prog sofimsha
head geometria
syst 3d
echo full
```

```
$ wezly definicja
```

```
let#nw 101
```

```
let#y -1.5
```

```
loop 7
```

```
node (#nw #nw+40 1) (0 0.5) #y 0
```

```
let#nw #nw+100
```

```
let#y #y+0.5
```

```
endloop
```

```
$ elementy powlokowe definicja
```

```
let#nw 101
```

Generowanie numeracji i współrzędnych węzłów. Zmienna *#nw* (identyfikująca szeregi węzłów, tu: 101, 201, 301, ..., 701), początkowa wartość 101. Zmienna *#y* (współrzędna globalna *Y*), początkowa wartość $Y = -1,5$ m, odpowiadająca węzłowi 101.

Otwarcie/ początek pętli, tu po siedmiu szeregach węzłów.

Generacja szeregu węzłów. W przejściu pierwszym pętli będą to węzły od 101 do 141 w odstępnie $\Delta X = 0,5$ m, poczynając od $X = 0$ m, przy stałym $Y = -1,5$ m i $Z = 0$ m.

Aktualizacja zmiennej *#nw*, tu o przyrost 100 w kolejnym cyklu pętli $nw := nw + 100$ (201, 301, ..., 701).

Aktualizacja zmiennej *#y*, tu o przyrost $\Delta Y = 0,5$ m w kolejnym cyklu pętli $Y := Y + 0,5$ (-1, -0,5, ..., +1,5).

Koniec pętli do definicji węzłów.

Generowanie elementów powłokowych.

Zmienna *#nw* (identyfikująca nr pierwszego węzła w elemencie, tu będzie to szereg: 101, 201, 301, ..., 601), początkowa wartość 101. Zobacz uwagi z rozdziału 4.1 dotyczące powiązania lokalnej numeracji węzłów elementu definiującej jego lokalny układ współrzędnych.

```

let#ne 1101
loop 6
  quad (#ne #ne+39 1) (#nw 1) (#nw+100 1) (#nw+101 1) (#nw+1 1)
  mno 2 t 250
endloop
$ elementy prętowe definicja
beam (2101 2140 1) (401 1) (402 1) ncs 1 np -1
$ podpory
node 401 - - - fix ppmx
      441 - - - fix xp
end

```

Zmienna *#ne* (identyfikująca szeregi elementów, tu: 1101, 1201, ..., 1601), początkowa wartość 1101.

Otwarcie/ początek pętli, tu po sześciu szeregach elementów.

(Uwaga. Przepisując kod do modułu *Teddy*, komendę *quad* zapisać w jednym wierszu!)

Generacja szeregu elementów. W cyklu pierwszym pętli będą to elementy od 1101 do 1140 z przyrostem 1. Należy zwrócić szczególną uwagę na incydencję, tj. funkcje *n1*, *n2*, *n3* i *n4* (rys. 4.4), definiującą orientację lokalnego układu współrzędnych elementu. Tu elementowi pierwszemu o nr. 1101 są przyporządkowane węzły w kolejności: 101, 201, 202, 102, a następnemu elementowi z tego szeregu o nr. 1102 – węzły: 102, 202, 203, 103 z przyrostem 1 itd.

Aktualizacja zmiennej *#nw*, tu o przyrost 100 w kolejnym cyklu pętli *nw := nw +100* (201, 301, ..., 601).

Aktualizacja zmiennej *#ne*, tu o przyrost 100 w kolejnym cyklu pętli *ne := ne +100* (1201, 1301, ..., 1601).

Koniec pętli do definicji elementów.

Generowanie elementów prętowych/ belkowych.

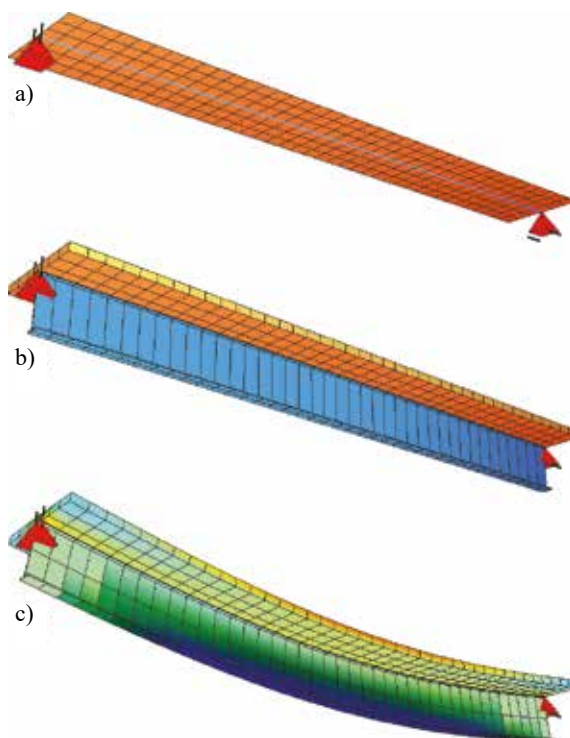
Generacja szeregu elementów od nr. 2101 do nr. 2140 z przyrostem 1. Tu elementowi pierwszemu o nr. 2101 są przyporządkowane węzły: 401, 402, a elementowi następnemu z tego szeregu o nr. 2102 – węzły: 402, 403 z przyrostem 1 itd. W zapisie pojawia się funkcja *np* o wartości -1. Dzięki niej program jako punkt zaczepienia przekroju przyjmie początek jego układu lokalnego, dodatkowo zastosuje mimośród definiowany w komendzie *sect* programu *AQUA*. UWAGA. Wartości dodatnie funkcji *np* dotyczą numeru profilu podłoża sprężystego. W przykładzie ta opcja nie jest używana.

Definicja warunków brzegowych (podpór).

```
$ ciezar wlasny
+prog ase
head cw
lc 1 dlz 1 titl cw
end
```

Obliczenia dla obciążenia ciężarem własnym.

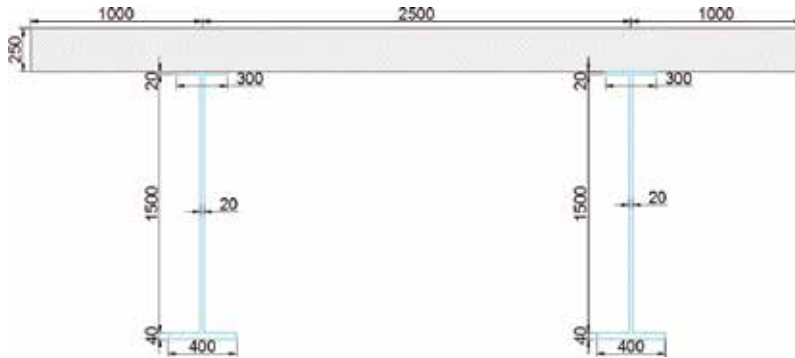
Na rys. 5.5, jako wyniki realizacji kodu danych do przykładu 5.1, przedstawiono kolejno: rzeczywisty schemat statyczny i jego wizualizację oraz deformację układu pod ciężarem własnym analizowanej belki zespolonej, uzyskane z poziomu modułu *Animator* 🎨.



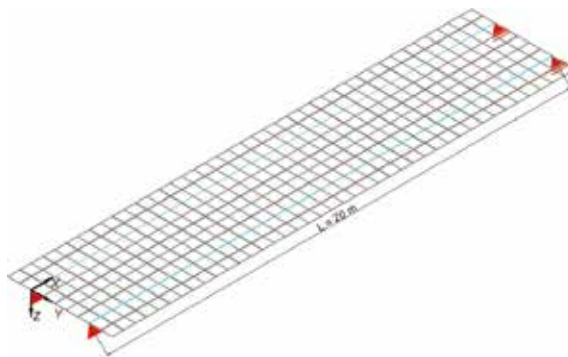
Rys. 5.5. Moduł *Animator*, belka zespolona; schemat statyczny (a), wizualizacja (b) oraz deformacja pod ciężarem własnym (c)

Zadanie 3. Dwudźwigarowa konstrukcja zespolona, *SOFIMSHA*

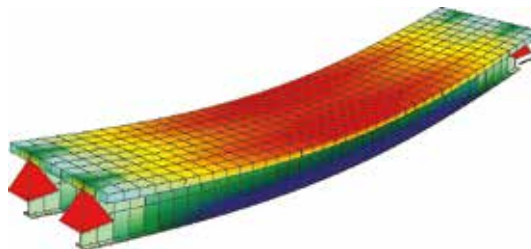
W ramach treningu rozbudować kod danych belki z przykładu 5.1 do konstrukcji zespolonej dwudźwigarowej wg rys. 5.6 oraz rys. 5.7 przy wykorzystaniu modułu *Teddy* 🧩 oraz programu *SOFIMSHA*. Następnie wykonać obliczenia od ciężaru własnego. Oczekiwane wyniki przedstawiono w formie wizualizacji, wykorzystując moduł *Animator* 🎨 (rys. 5.8).




Rys. 5.6. Przekrój poprzeczny konstrukcji zespolonej




Rys. 5.7. Schemat statyczny konstrukcji zespolonej

Rys. 5.8. Moduł *Animator* – wizualizacja konstrukcji zespolonej pod obciążeniem ciężarem własnym


5.2. Definiowanie geometrii – SOFIMSHC

W dotychczasowych przykładach geometrie rozważanych ustrojów były tworzone z wykorzystaniem programu *SOFIMSHA* w module *Teddy* . Sposób ten wymaga samodzielnego opracowania i budowy struktury dyskrytacyjnej (węzłów oraz elementów), umożliwia jednak pełną kontrolę nad tworzoną siatką elementów skończonych. Niestety w przypadku obiektów skomplikowanych geometrycznie postępowanie takie może być bardzo uciążliwe, a czasem wręcz niewykonalne w rozsądnym czasie. W przypadkach skomplikowanych można

wykorzystać alternatywną technikę oferowaną przez program *SOFIMSHC*, tzw. modelowanie geometryczne. Program *SOFIMSHC* jest również dostępny z poziomu modułu *Teddy* .

W tzw. modelowaniu geometrycznym, zamiast bezpośredniego definiowania węzłów i elementów siatki dyskredytacyjnej, buduje się geometrię układu z charakterystycznych dla niego obiektów w postaci: punktów, linii, powierzchni i brył. Tak wyróżnionym obiektom geometrycznym można przypisać różne charakterystyki konstrukcyjne i fizyczne, np. punktom – więzy, liniom – przekroje i materiały, powierzchniom – grubości i materiały itd. Na podstawie tych informacji oraz ustalonych zakresów parametrów sterujących kryteriami dyskredytacyjnymi program automatycznie generuje siatki tworzące model obliczeniowy układu.

Przykład 5.2. Zachowując parametry techniczne ustroju z przykładu 5.1 (rys. 5.1), zmodyfikować utworzony tam kod danych, dostosowując go do wymagań programu *SOFIMSHC*.

W zakresie definiowania materiałów z poziomu modułu *Teddy*  należy w kodzie do programu *AQUA* z poprzedniego zadania wprowadzić tylko jedną zmianę. Należy wyzerować odpowiedzialny za dystans/ mimośród parametr funkcji *zm* w komendzie *sect*. Modyfikacja ta wynika z innego sposobu uwzględniania mimośrodków w programie *SOFIMSHC*.

```
+prog aqua
head przekroje+materiały
echo full
norm en 1993-2005 00 unit 5 $ norma
$ material
stee 1 s 355 titl s355 $ stal
conc 2 c 35 titl c35/45 $ beton
$ przekroj poprzeczny
sect 1 mno 1 zm 0 titl przekroj_a
poly type opz
vert 1 0 0
      2 150 0
      3 150 20
      4 10 20
      5 10 1520
      6 200 1520
      7 200 1560
      8 0 1560
end
```

```
+prog sofimshc
head geometria
syst 3d
```

```
ctrl mesh 2+64; ctrl hmin 0.5
```

Wywołanie programu *SOFIMSHC*.

W komendzie *syst* oznaczenia i ich znaczenia są identyczne jak w *SOFIMSHA* (rys. 3.10).

Komenda *ctrl* ustala parametry sterujące podziałem na elementy skończone (rys. 5.10).

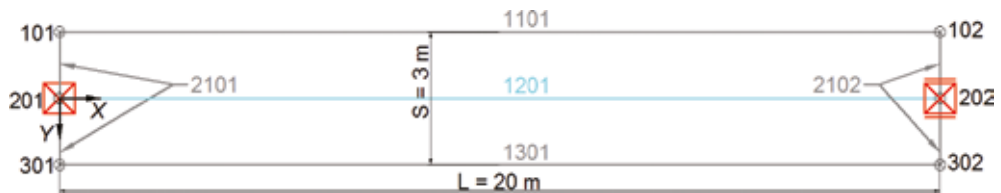
Opcja *mesh 2* wybiera tworzenie elementów prętowych i powierzchniowych.

Opcja *mesh +64* wymusza tworzenie czterowęzłowych elementów typu *quad*.

Opcja *hmin* 0.5 ustala maksymalny rozmiar elementu, który tu zadeklarowano na 0,5 m. Wartość tę można dalej zmienić indywidualnie.

UWAGA. Średnik jako separator komend pozwala zapisywać różne komendy (szczególnie krótkie) w jednym wierszu bez konieczności zmiany linii.

Przed przystąpieniem do wprowadzania obiektów geometrycznych należy ustalić ich oznaczenia/ numerację (rys. 5.9). Można sobie tutaj pozwolić na pewną dowolność: wprowadzane punkty mogą się znajdować wewnątrz linii lub powierzchni, linie mogą się krzyżować bez oznaczenia wspólnego punktu, ewentualnie mogą się zaczynać lub kończyć w środku innej linii.



Rys. 5.9. Modelowanie geometryczne belki zespolonej, oznaczenia/ numeracja punktów i linii, widok z góry

CTRL			
Item	Description	Unit	Default
OPT	A literal from the following list:	LIT	!
	TOPO Topological decomposition	LIT/-	ON
	ON	-	0
	OFF	-	-
	DEL	-	-
	GAXP	-	+3
	SARB	-	+8
	XFLG	-	+1
	TOLG Intersection tolerance	[m] ₁₀₀₁	-0.01
	NODE Start index of automatically assigned numbers	-	1000
	DELN Deletion of unused elements	-	1
	HEAL Geometry healing	LIT	!
	JOIN	[m] ₁₀₀₁	1.0
	DELO	[m ²] ₁₀₀₂	1.0
	MESH Start of mesh generation	-	0
	HMIN Mesh size	[m] ₁₀₀₁	1.0
	FINE Refinement at nodes	[m] ₁₀₀₁	HMIN
	EFAC Refinement at short edges	-	1.4
	PROG Progression factor	-	1.5
	BSEC Generation of design sections	-	0
	LOCA Local coordinate systems	-	1
	TOLN geom. tolerance detection of identical FE-nodes	-	1.e-6

Odczytanie modelu geometrycznego, uwzględnienie przecięć elementów składowych, stworzenie spójnego mechanicznie układu strukturalnego

Tolerancja dla podziałów, np. dwa punkty w odległości mniejszej niż wartość *TOLG* na etapie generacji siatki zostaną połączone w jeden.

Wartość początkowa dla numerów automatycznie wygenerowanych węzłów

Opcje generatora siatek, np:
0 – brak generacji
1 – podział na elementy typu *beam*
2 – podział na elementy typu *beam* oraz *quad*

Rozmiar siatki dla całego układu

Rys. 5.10. Pomoc/ instrukcja do programu *SOFIMSHC* – wybrane funkcje komendy *ctrl*

\$ punkty

```
spt 101 0 -1.5 0
      102 20 -1.5 0
      201 0 0 0 fix ppmx
      202 20 0 0 fix xp
      301 0 1.5 0
      302 20 1.5 0
```

Komenda *spt* definiuje punkty (rys. 5.11).

Funkcją *fix* można przypisać od razu warunki brzegowe nałożone na punkt.

\$ linie wzdluzne

```
sln 1101 101 102
      1201 201 202 sno 1 eza 0.9192+0.25/2 eze 0.9192+0.25/2
```

Komenda *sln* definiuje linię *sln* (rys. 5.12).

Funkcja *sno* przypisuje linii przekrój pręta, tu nr 1, linia jest traktowana jako oś pręta; jest to podstawa generacji siatki elementów prętowych.

Funkcje *eza* i *eze* przypisują linii/ osi mimośród, tu na kierunku osi (*Z*), który może być inny w węzle początkowym (*a*) i końcowym (*e*). Tutaj mimośród, zgodnie z definicją, jest liczony od środka ciężkości przekroju, a jego wartość $0,9192 + 0,25/2$ przyjęto na podstawie wyników obliczeń z programu *AQUA* (zob. tabela na rys. 3.7).

```
1301 301 302
```

\$ linie poprzeczne

```
sln 2101 101 301
      2102 102 302
```

SPT			
Item	Description	Unit	Default
NO	Point number	-	-
X	Point coordinates	[m] ₁₀₀₁	0.0
Y		[m] ₁₀₀₁	0.0
Z		[m] ₁₀₀₁	0.0
REF	Relative to / projection on PT, PCYL, PPOL AX, AXYZ, AXZX, AXXY AR, ARYZ, ARZX, ARXY AXAX, AXAR	LIT	-
NREF	Identifier of projection target	-/Lit4	-
FIX	Support condition	Lit16	-
BX	Width of support / mesh in local x	[m] ₁₀₀₁	-
BY	Width of support / mesh in local y	[m] ₁₀₀₁	-
T	Plate thickness if lying inside a SAR	[mm] ₁₀₁₀	-

Numer punktu

Współrzędne punktu

Możliwość przypisania podpory

Możliwość zdefiniowania grubości, jeśli punkt należy do powierzchni

Rys. 5.11. Pomoc/ instrukcja do programu *SOFIMSHC* – wybrane funkcje komendy *spt*

SLN			
Item	Description	Unit	Default
NO	Line number	-	-
NPA	Point at start of line	-	-
NPE	Point at end of line	-	-
REF	Reference to a geometry at GAX	-/LIT	-
FIX	Boundary conditions along the line	Lit16	-
SDIV	>0: Mesh density used for subdivision <0: Number of elements to be generated	[m]1001	-
GRP	Group number	-	-
STYP	Element type, subdivision	LIT	SE
SNO	Number of cross section	-/Lit12	-
NP	Number of bore / bedding profile	-	-
KR	Direction identifier or	LIT	-
DRX	explicit direction vector of the local	-	0.
DRY	beam coordinate system	-	0.
DRZ		-	0.
DROT	Additional rotation about beam axis	deg	0.
EXA	Eccentricity in global coordinates	[m]1001	0.
EYA	at start	[m]1001	0.
EZA		[m]1001	0.
EXE	Eccentricity in global coordinates	[m]1001	0.
EYE	at end	[m]1001	0.
EZE		[m]1001	0.
FIXA	Hinge conditions at start	-/Lit16	-
FIXE	Hinge conditions at end	-/Lit16	-

Rys. 5.12. Pomoc/ instrukcja do programu *SOFIMSHC* – wybrane funkcje komendy *sln*

\$ powierzchnia

```
sar 1 mno 2 t 250 qref cent mctl 1
```

Komenda *sar* definiuje powierzchnię. Można jej przypisać: materiał, grubość, mimośród (rys. 5.13). Na tej podstawie generuje się elementy powierzchniowe.

Funkcja *mctl* o wartości 1 wybiera siatkę regularną (tzw. strukturalną), jeśli spełnione są do tego warunki.

```
sarb type out nl 1101,1301,2101,2102
```

Komenda *sarb* ogranicza liniami daną powierzchnię (rys. 5.14). Opcja *out* funkcji *type* wskazuje, że linie te są obrysem zewnętrznym.

```
end
```

```
+prog ase
```

```
head cw
```

```
lc 1 dlz 1 titl cw
```

```
end
```

Obliczenia dla ciężaru własnego.

SAR			
Item	Description	Unit	Default
NO	Area number/ Type identifier PROP attribute region VOID opening	-/LIT	0
FIX	Boundary condition within the area	LIT8	*
GRP	Group number of elements in area	-	0
MNO	Material number of QUADs in area	-	0
MRF	Material number of reinforcement	-	0
REF	Geometric reference on area SAR	LIT4	*
NX	Vector defining the upward direction of the area	-	*
NY		-	*
NZ		-	*
NRA	Formulation of elements (Bitmask) 0 geometry only +1 plate action +2 membrane action +4 in-plane bending	-	0/7
QREF	Reference for elements CENT centered ABOV in negative z direction BELO in positive z direction	LIT	CENT
KR	Direction identifier or	LIT/-	*
DRX	explicit direction vector of the local	-	-
DRY	element coordinate system	-	-
DRZ	see remarks	-	-
DROT	Rotation angle	[deg]	0.0
T	Constant thickness override	[mm]1010	*
MCTL	Meshing control AUTO Automatic meshing REGM Regular Mesh if possible SNGQ Mesh as single quad OFF Deactivate meshing		0
	Mesh density:		
H1	- for the whole region	[m]1001	-
H2	- at points inside the region	[m]1001	-
H3	Minimum thickness of elements within the structured boundary layer mesh defined with DFIX (- SARB)	[m]1001	-
XFLG	Disable automatic intersection	LIT	-
TITL	Title of region	Lit32	-

Rys. 5.13. Pomoc/ instrukcja do programu SOFIMSHC – wybrane funkcje komendy sar

Wynik realizacji powyższego kodu przedstawiono w postaci wizualizacji w module *Animator* (rys. 5.15a). Wskazuje ona, że kontrola nad parametrami siatki nie jest tu dostatecznie skuteczna, tak jak ma to miejsce w tzw. modelowaniu strukturalnym z użyciem modułu SOFIMSHA. Bowiem na rys. 5.15a, w poprzek płyty są cztery, a po jej długości dwadzieścia

elementów, co daje ich wymiar $0,75 \times 0,5$ m mimo deklaracji $h_{min} = 0,5$ m w komendzie *ctrl*. Niemniej jednak elastyczność zmian gęstości siatki w tzw. modelowaniu geometrycznym jest nieporównywalnie wyższa, wystarczy bowiem zmienić jeden parametr $h_{min} = 0,25$ m i otrzymuje się model o zagęszczonym podziale (rys. 5.15b). Tym razem elementy mają wymiary dokładnie $0,25 \times 0,25$ m.

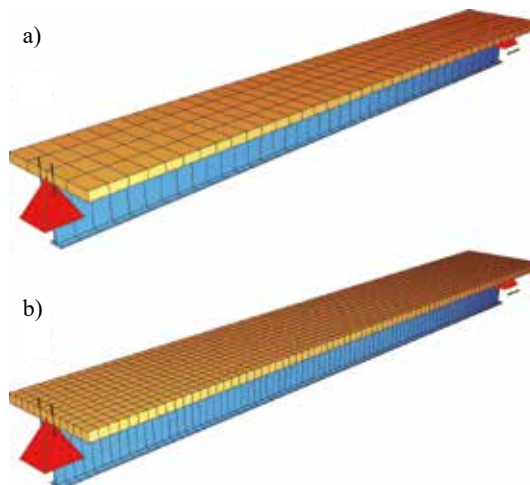
SARB			
Item	Description	Unit	Default
TYPE	Type of boundary	LIT	OUT
	OUT outer boundary		
	IN inner boundary		
	CONS inner constraint		
NL	Number of a geometric line	-	0
NP	Number of a geometric point	-	0
NA	number of start point	-	*
NE	number of end point	-	*
T	Thickness of the area in this boundary	[mm] ₁₀₁₀	*
MNO	Material number of the boundary	-	0
FIX	Hinge condition of boundary	LIT	-
DFIX	Distance to boundary	m	0
CA	Axial bedding	[kN/m ²] ₁₀₀₆	0
CL	Lateral bedding	[kN/m ²] ₁₀₀₆	0
CD	Torsional bedding	[kNm/m] ₁₀₀₅	0

Rodzaj obiektu, do którego ma zostać przyłożone obciążenie

Wartość marginesu – zakresu obszaru przeszukiwania

Typ obciążenia

Rys. 5.14. Pomoc/ instrukcja do programu *SOFIMSHC* – wybrane funkcje komendy *sarb*



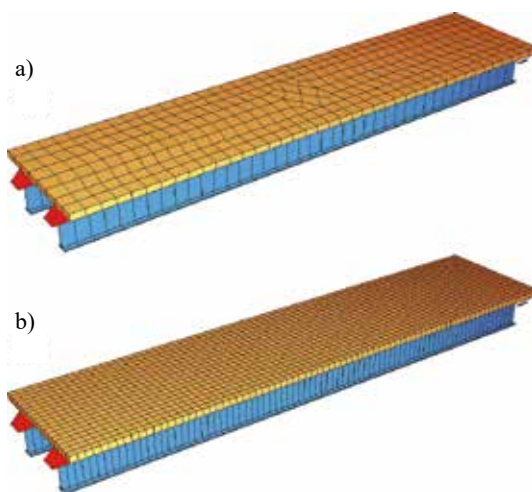
Rys. 5.15. Moduł *Animator* – wizualizacja belki zespolonej, wymiar elementu $h_{min} = 0,5$ m (a), $h_{min} = 0,25$ m (b)

Wynik ten wskazuje, że w przypadku generowania siatek techniką modelowania geometrycznego konieczne jest ich wizualne kontrolowanie, czy oczekiwany wynik podziału został

osiągnięty i nie pojawiają się obszary o niewłaściwym układzie elementów o zachwianych proporcjach kształtu!

Zadanie 4. Dwudźwigarowa konstrukcja zespolona, SOFIMSHC

W ramach treningu stworzyć ponownie kod danych wg wytycznych do zadania 3 (rys. 5.6, rys. 5.7), wykorzystać moduł *Teddy* oraz program *SOFIMSHC*. Wykonać obliczenia układu od oddziaływania ciężarem własnym. Oczekiwany efekt przedstawiono jako wizualizację w module *Animator* (rys. 5.16). Stosując parametr $h_{min} = 0,5$ m, nie uzyskuje się mimo deklaracji siatki regularnej, występuje zachwianie jej struktury (rys. 5.16a); sytuacja poprawia się po przyjęciu parametru $h_{min} = 0,25$ m (rys. 5.16b).





Rys. 5.16. Moduł *Animator*– wizualizacja konstrukcji zespolonej, parametr wymiarowy elementu: $h_{min} = 0,5$ m (a), $h_{min} = 0,25$ m (b)




Należy podkreślić, że wybór parametru h_{min} , sterującego maksymalnym uogólnionym wymiarem elementu skończonego, a stąd gęstością i regularnością generowanej automatycznie siatki podziału, nie powinien być przypadkowy czy dowolny. Określenie wartości parametru h_{min} należy starannie przemyśleć. Mimo fizycznego wymiaru h_{min} , tj. długości [m], należy go przede wszystkim powiązać z planowaną liczbą podziału na elementy skończone charakterystycznych wymiarów konstrukcji, szczególnie tych minimalnych. Ten podział będzie bowiem decydował o zbieżności rozwiązań MES, a więc o ich jakości i wiarygodności. Z perspektywy dyskretyzacji charakterystycznymi wymiarami konstrukcji są nie tylko jej oczywiste wymiary gabarytowe, takie jak długość, szerokość i wysokość, ale także podziały na kondygnacje, wysokości wsporników itp. czy też rozstawy: wzmocnień, żeber, krawędzi załamań i krzyżowań, zmiany grubości itp.

Należy także podkreślić – z punktu widzenia jakości i wiarygodności rozwiązań oraz możliwości ich kontrolowania – przewagę siatek regularnych nad podziałem nieregularnym. W przypadku siatek regularnych, inaczej strukturalnych, mamy pełną kontrolę nad kształtem i wymiarami/proporcjami elementu skończonego wynikającymi z określonej struktury oczek

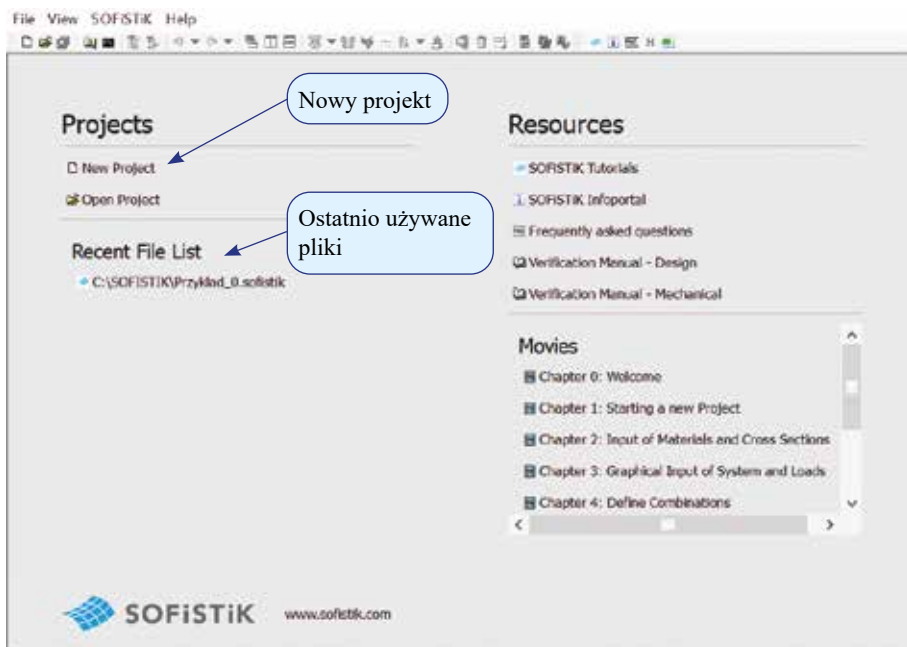
siatki. W przypadku siatek nieregularnych, inaczej niestrukturalnych, może się pojawić – związany z nieregularnością podziału – problem odporności poszczególnych typów elementów skończonych na dystorsję ich kształtu. Z teorii MES wiadomo bowiem, że ani kształt, ani proporcje elementów skończonych, ani także sposób ich incydencji nie mogą być dowolne (zob. np. RACHOWICZ [2012]).

5.3. Materiały, przekroje, geometria – SOFIPLUS

Kolejnym, bardzo atrakcyjnym sposobem definiowania tzw. modelu obliczeniowego w programie SOFiSTiK jest moduł *SOFIPLUS-X* . Bazuje on na tzw. modelowaniu geometrycznym, czyli wykorzystuje obiekty takie jak: punkty, linie, powierzchnie oraz bryły, którym przypisuje się odpowiednie charakterystyki konstrukcyjne i fizyczne (faktycznie wykorzystując program *SOFIMSHC*). Sposób pracy z użyciem modułu *SOFIPLUS-X*  różni się jednak tym, że wszystko odbywa się nie tak jak wcześniej, poprzez zapis kodu tekstowego, ale poprzez środowisko graficzne CAD. Dlatego, aby skorzystać z tego podejścia, niezbędna jest podstawowa znajomość narzędzi programu AutoCAD.

Rozpoznanie modułu *SOFIPLUS-X*  będzie prowadzone na danych belki zespolonej z przykładów 5.1 i 5.2 (rozdziały 5.1, 5.2). Do tego celu zostanie wykorzystane środowisko *SOFiSTiK Structural Desktop*  (*SSD*) zapewniające interakcję z innymi modułami, między innymi również *Teddy* .

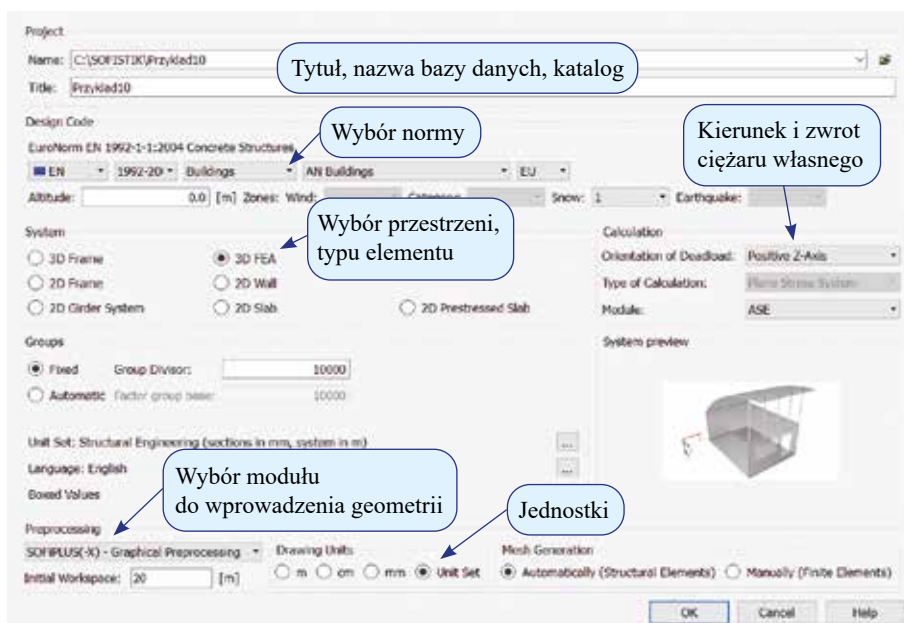
Po otwarciu modułu *SSD*  jako pierwsze pojawia się okno startowe (rys. 5.17).



Rys. 5.17. Moduł *SSD* – okno startowe programu

W tym oknie należy kliknąć ikonę *Nowy projekt (New Project)*. Wtedy następuje przejście do okna, w którym ustala się nazwę projektu, katalog zapisu oraz należy dokonać wyboru opcji ogólnych (rys. 5.18). Na potrzeby zadania ustalono: normy *EN*, pracę w przestrzeni *3D FEA* (model przestrzenny oraz wykorzystanie elementów prętowych, powierzchniowych lub bryłowych o maksymalnej dopuszczalnej liczbie stopni swobody), kierunek grawitacji zgodny z osią *Z* układu globalnego oraz moduł *SOFIPLUS-X* jako narzędzie do tworzenia geometrii. Dokonane ustawienia należy zatwierdzić przez ikonę *OK*.

Kolejnym oknem jest docelowy moduł *SSD*. Najwygodniej posługiwać się nim, korzystając z drzewa wyboru umieszczonego po jego lewej stronie (rys. 5.19).

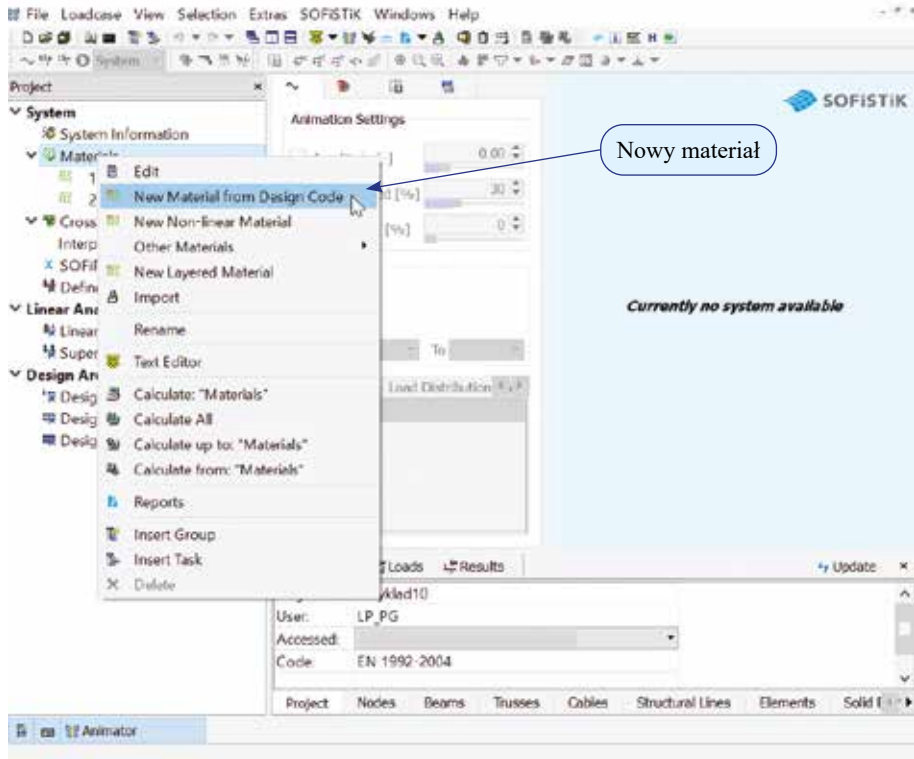


Rys. 5.18. Moduł *SSD* – okno wyboru opcji ogólnych

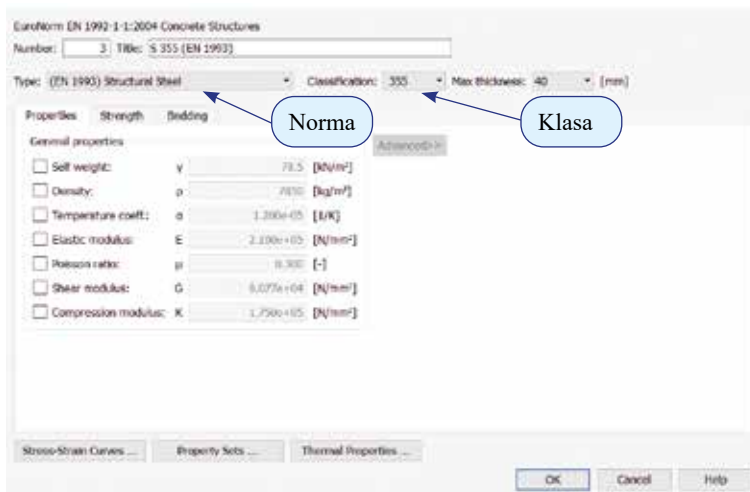
Na tym etapie istnieje już możliwość zdefiniowania materiałów. Wystarczy kliknąć prawym klawiszem *Materiały (Materials)*, a następnie wybrać opcję *Nowy (New)*. Korzystając z bazy danych (norma: *EN 1993* – stale, *EN 1992* – betony), należy utworzyć dwa materiały: stal *S 355* (rys. 5.20) oraz beton *C 35/45* (rys. 5.21).

Pozostałe etapy budowy modelu obliczeniowego: przekroje, geometria oraz obciążenia będą realizowane w module *SOFIPLUS-X*. Otwarcie tego programu następuje po dwukrotnym kliknięciu lewym klawiszem myszy na jego nazwę w drzewku wyboru (rys. 5.22).

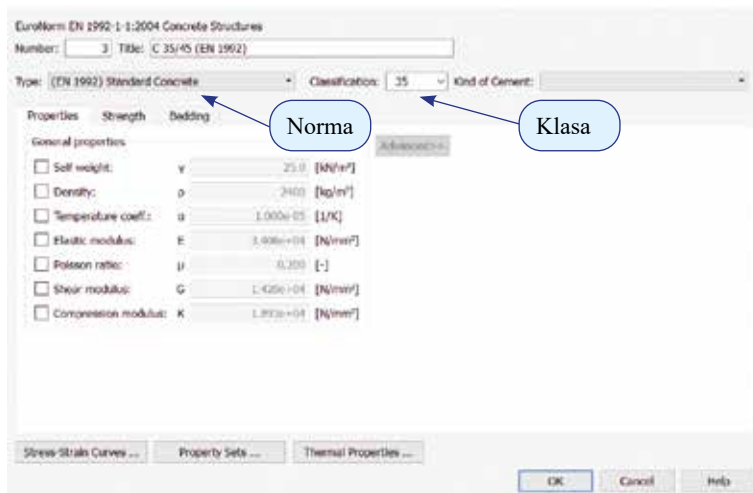
Ekran modułu *SOFIPLUS-X* przedstawiono na rys. 5.23. Jest on zbliżony do typowego ekranu programu *AutoCAD*. Różni się głównie dodatkowym oknem po lewej stronie, w którym w kilku zakładkach umieszczono drzewa rozwijalne do kolejnych etapów tworzenia modelu obliczeniowego. W zakładce pierwszej *System (System)* istnieje możliwość definiowania lub edycji m.in. materiałów oraz przekrojów. Tutaj dane materiałów, które zostały utworzone wcześniej, zostały zaimportowane z bazy danych (materiał nr 3 i 4).



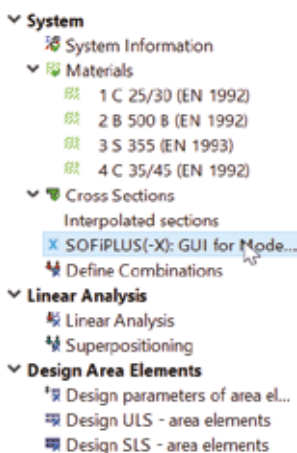
Rys. 5.19. Moduł SSD – nowy materiał



Rys. 5.20. Moduł SSD – nowy materiał – stal S 355



Rys. 5.21. Moduł SSD – nowy materiał – beton C 35/45



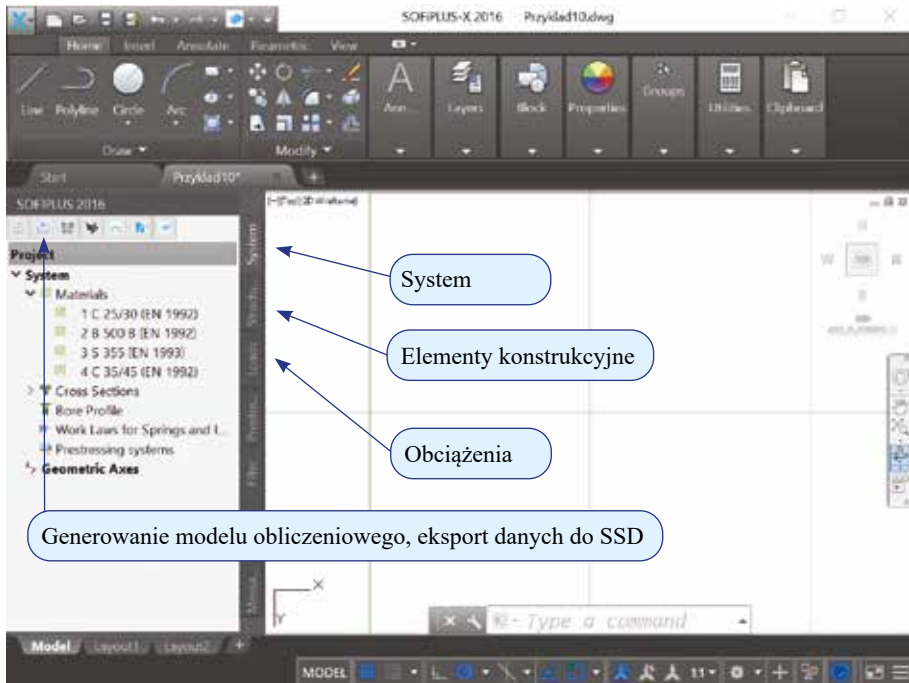
Rys. 5.22. Moduł SSD – otwarcie modułu SOFIPLUS-X

Kolejnym etapem będzie opis blachownicy stalowej. Uruchomienie opcji tworzenia nowego przekroju (rys. 5.24) wymaga kliknięcia prawym klawiszem *Przekroje* (*Cross Sections*), a następnie należy wybrać *Przekrój nowy lity* (*New Solid Section*) → *Ogólny* (*General*). Oczywiście można zastosować również inną wersję, np. cienkościenny; tu został zachowany sposób wykorzystywany w dotychczas zrealizowanych ćwiczeniach.

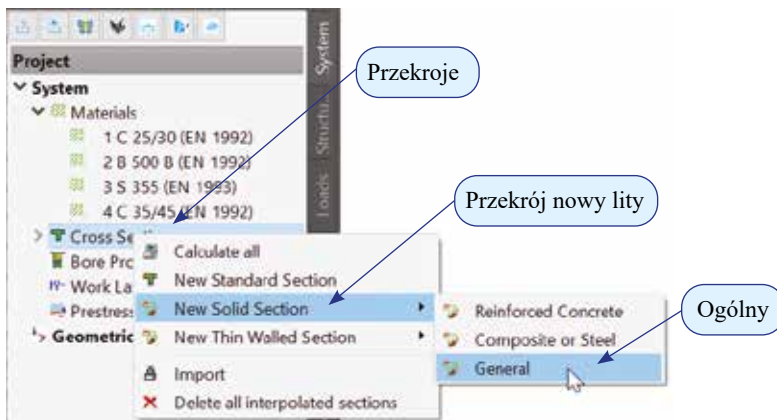
Następuje otwarcie okna *Parametry przekroju* (*Section Properties*), w którym ustawiamy materiał główny jako nr 3 (S 355) oraz typ elementu belka (rys. 5.25). Wybór zatwierdzamy *OK*.

W kolejnym oknie pojawia się edytor przekrojów (rys. 5.26). Za jego pomocą, korzystając z tradycyjnych narzędzi AutoCAD, można narysować blachownicę. Należy zwrócić

uwagę, że ewentualny mimośród trzeba uwzględnić już na tym etapie. Dlatego górną krawędź przekroju odsunięto od początku układu współrzędnych o połowę grubości płyty betonowej $250/2 = 125$ mm.



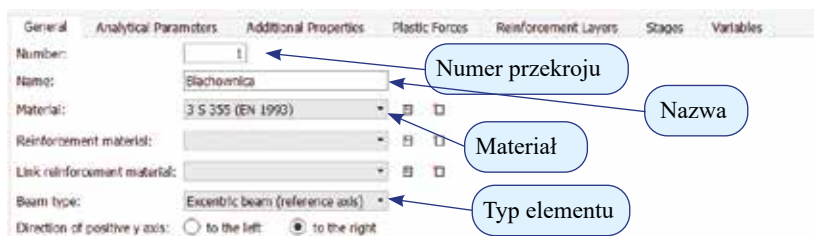
Rys. 5.23. Moduł *SOFIPLUS-X* – ekran pracy programu



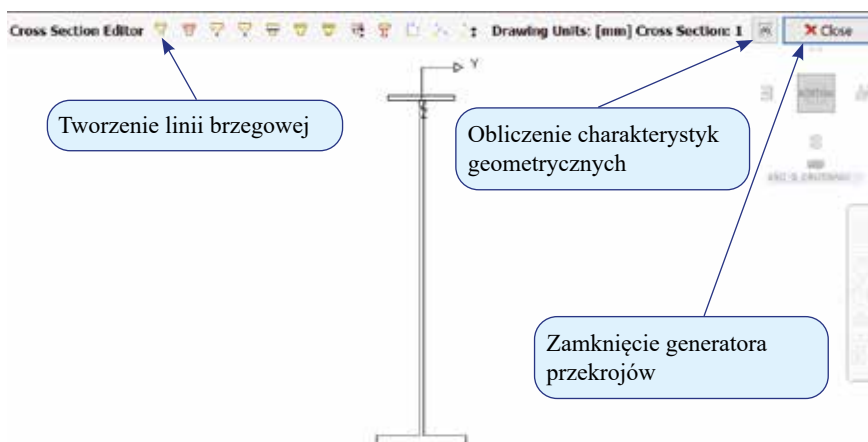
Rys. 5.24. Moduł *SOFIPLUS-X* – tworzenie nowego przekroju

W kolejnym kroku, stosując narzędzie *Tworzenie linii brzegowej (Creates a boundary line for a solid cross section...)*, należy wskazać linie tworzące przekrój. Wybór linii (domyślnie wyszukiwane są punkty) rozpoczyna się kliknięciem prawym klawiszem myszy w dowol-

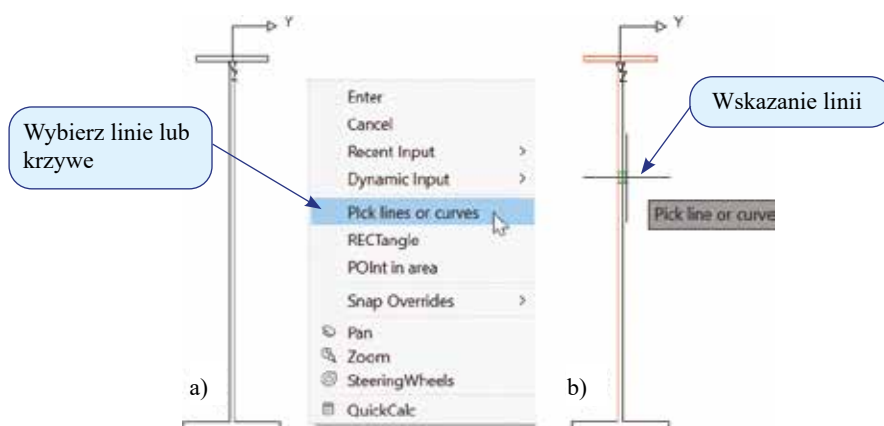
nym punkcie okna graficznego, a następnie należy zmienić tryb na *Wybierz linie lub krzywe* (*Pick lines or curves*) (rys. 5.27a). Po tym zabiegu wystarczy kliknąć w każdą z linii tworzących blachownicę; powinny one zmieniać kolor na czerwony (rys. 5.27b).




Rys. 5.25. Moduł *SOFIPLUS-X* – parametry przekroju

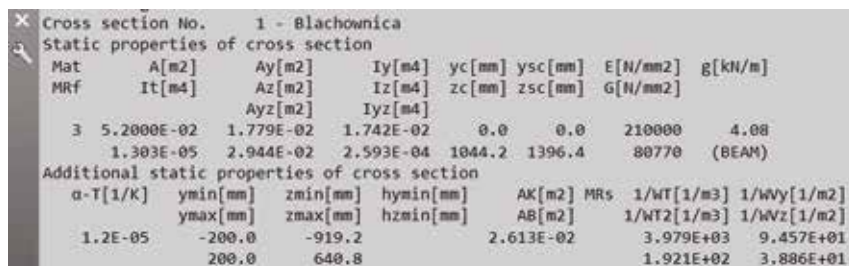


Rys. 5.26. Moduł *SOFIPLUS-X* – edytor przekrojów



Rys. 5.27. Moduł *SOFIPLUS-X* – zmiana trybu pracy kursora (a), wybór linii (b)

Po wybraniu wszystkich linii (zamknięty obwód) przekrój automatycznie zostaje przyjęty, jego narożniki przyjmują formę kropek. Narzędzie *Tworzenie linii brzegowej* (*Creates a boundary line for a solid cross section...*) można zamknąć (klikając X okna systemu Windows). Natychmiast można sprawdzić parametry statyczne, klikając obliczenia charakterystyk geometrycznych  (rys. 5.26). Wynik pojawia się w oknie tekstowym poniżej okna graficznego (rys. 5.28).



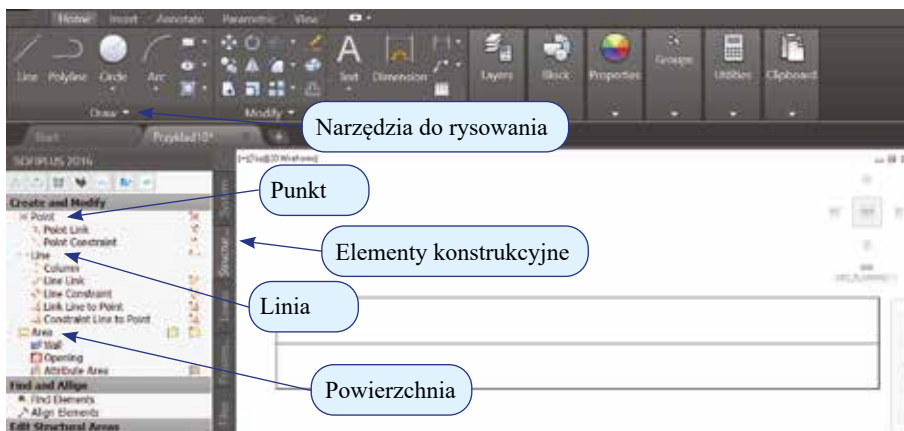
Cross section No. 1 - Blachownica							
Static properties of cross section							
Mat	A[m2]	Ay[m2]	Iy[m4]	yc[mm]	ysc[mm]	E[N/mm2]	g[kN/m]
MRF	It[m4]	Az[m2]	Iz[m4]	zc[mm]	zsc[mm]	G[N/mm2]	
3	5.2000E-02	1.779E-02	1.742E-02	0.0	0.0	210000	4.08
	1.303E-05	2.944E-02	2.593E-04	1044.2	1396.4	80770	(BEAM)
Additional static properties of cross section							
alpha-T[1/K]	ymin[mm]	zmin[mm]	hymin[mm]	AK[m2]	MRs	1/Wt1[1/m3]	1/WVy[1/m2]
	ymax[mm]	zmax[mm]	hzmin[mm]	AB[m2]		1/Wt2[1/m3]	1/WVz[1/m2]
1.2E-05	-200.0	-919.2		2.613E-02		3.979E+03	9.457E+01
	200.0	640.8				1.921E+02	3.886E+01

Rys. 5.28. Moduł *SOFIPLUS-X* – właściwości statyczne przekroju (w układzie lokalnym przekroju)

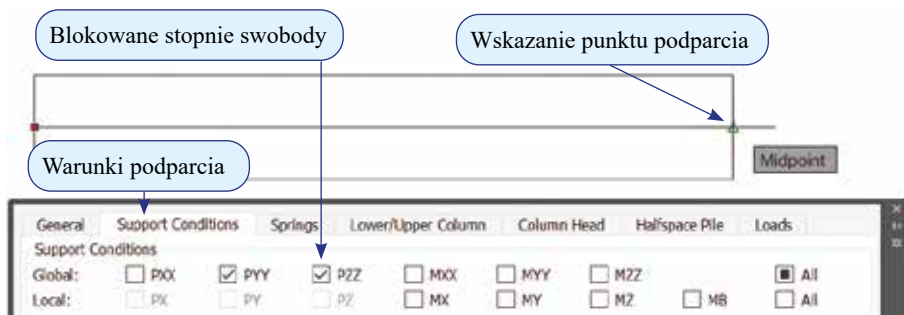
Pracę edytora przekrojów kończy się, klikając *Zamknij* (*Close*) (rys. 5.26) i zatwierdzając zapis. Przekrój został utworzony. W przypadku konieczności powrotu do edycji wystarczy dwa razy kliknąć w jego nazwę.

Kolejny etap pracy to stworzenie geometrii belki. Budowę należy rozpocząć od linii obrysu płyty oraz osi blachownicy w płaszczyźnie *XY* układu globalnego przy użyciu standardowych narzędzi AutoCAD (rys. 5.29). W następnym kroku obiektom graficznym trzeba przypisać charakterystyki elementów skończonych; czynność tę wykonuje się w zakładce *Elementy konstrukcyjne* (*Structural elements*) (rys. 5.23). Za pomocą narzędzia *Punkt* (*Point*) (rys. 5.29) możliwe jest wprowadzenie punktowych warunków brzegowych. W opcjach przedstawionych na rys. 5.30 blokuje się wybrane stopnie swobody (*PXX*, *PYY*, *PZZ*, *MXX*), następnie bez zamykania okna opcji przechodzi się do okna graficznego i wskazuje punkt, w którym podpora ma się pojawić. W przypadku zmiany warunków brzegowych w innych punktach wraca się do okna opcji, aktualizuje więzy na stopniach swobody (*PYY*, *PZZ*) i ponownie przechodzi się do okna graficznego. Jeśli wszystkie warunki brzegowe są już zdefiniowane, okno *Punkt konstrukcyjny* (*Structural Point*) można zamknąć. Wykorzystując kolejne narzędzie *Linia* (*Line*) (rys. 5.29), liniom nadaje się charakterystyki (*Structural line*) tworzące siatkę elementów prętowych (rys. 5.31).

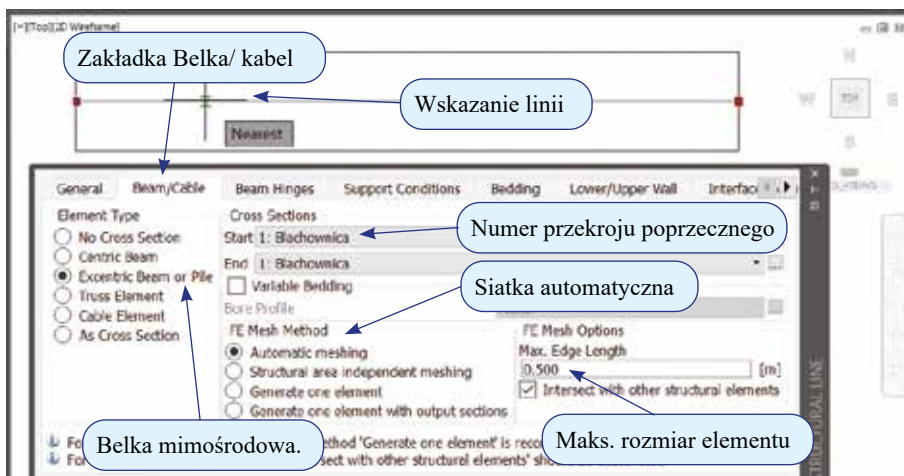
W oknie *Linia konstrukcyjna* (*Structural line*) należy zdefiniować parametry takie, jak: typ elementu, przekrój, sposób generacji siatki oraz maksymalny rozmiar elementu. Następnie bez zamykania okna opcji przechodzi się do okna graficznego i wskazuje linię, na podstawie której ma powstać siatka dyskretyzacyjna prętów. Tak jak w przypadku opisu przekroju, czynność tę należy poprzedzić zmianą trybu pracy kursora na *Wybierz linie lub krzywe* (*Pick lines or curves*) (rys. 5.27a). Wskazana linia powinna zmienić kolor na żółty. Na zakończenie wystarczy zamknąć okno *Linia konstrukcyjna* (*Structural line*).



Rys. 5.29. Moduł SOFIPLUS-X – rysowanie linii tworzących geometrię

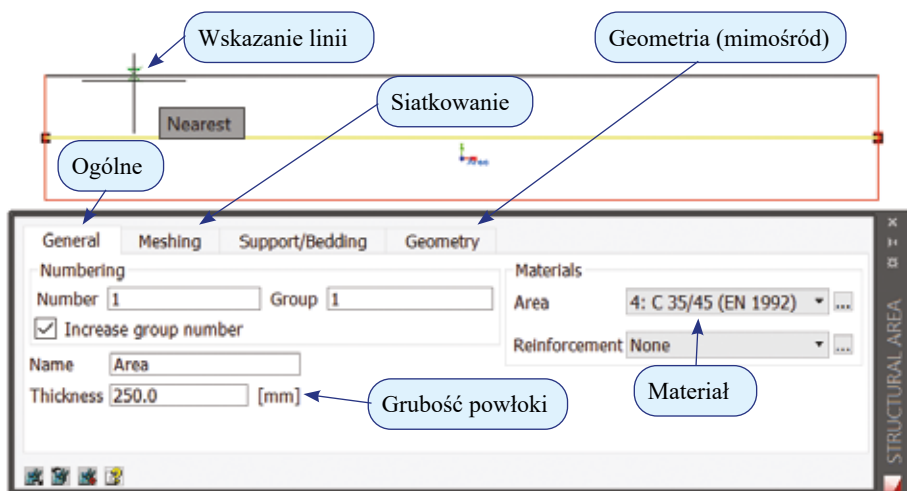


Rys. 5.30. Moduł SOFIPLUS-X – definicja podpory



Rys. 5.31. Moduł SOFIPLUS-X – tworzenie elementów prętowych

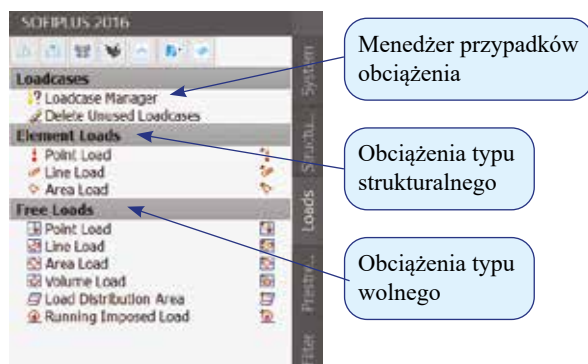
Ostatni etap tworzenia geometrii tego przykładu to wykorzystanie narzędzia *Powierzchnia* (*Area*) (rys. 5.29) do definiowania elementów powłokowych (rys. 5.32). W oknie *Powierzchnia konstrukcyjna* (*Structural Area*) w zakładce *Ogólne* (*General*) należy podać parametry takie jak grubość powłoki i materiał; w zakładce *Siatkowanie* (*Meshing*) należy zaznaczyć opcję generatora siatek *Twórz siatkę regularną* (*Create regular mesh*) i podać rozmiar elementów *Gęstość* (*Density*) 0,5 m oraz ewentualny mimośród (*Element Alignment*) w zakładce *Geometria* (*Geometry*). Następnie bez zamykania okna opcji przechodzi się do okna graficznego i wskazuje linie obwodowe obszaru. Również w tym wypadku czynność tę należy poprzedzić zmianą trybu pracy kursora na *Wybierz linie lub krzywe* (*Pick lines or curves*) (rys. 5.27a). Wskazane linie powinny zmienić kolor na czerwony, po zamknięciu obszaru/ obwodu – na niebieski. Na zakończenie zamyka się okno *Powierzchnia konstrukcyjna* (*Structural Area*).



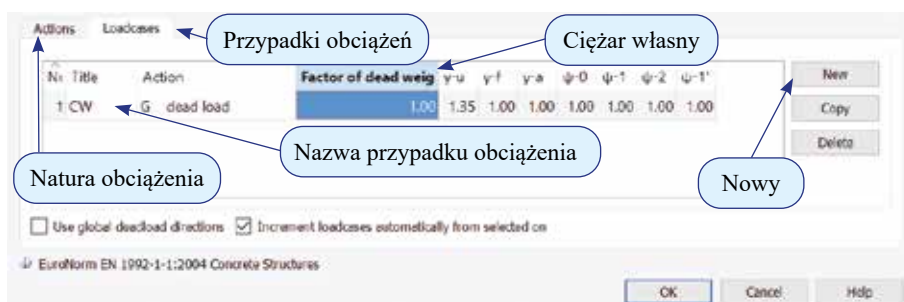
Rys. 5.32. Moduł *SOFIPLUS-X* – tworzenie elementów powierzchniowych

Ewentualną ponowną edycję obiektów/ elementów konstrukcji uruchamia się w oknie graficznym, klikając dwa razy w obiekt (punkt, linia, powierzchnia) lub jego nazwę. Wówczas automatycznie otwiera się okno z wprowadzonymi wcześniej parametrami obiektu.

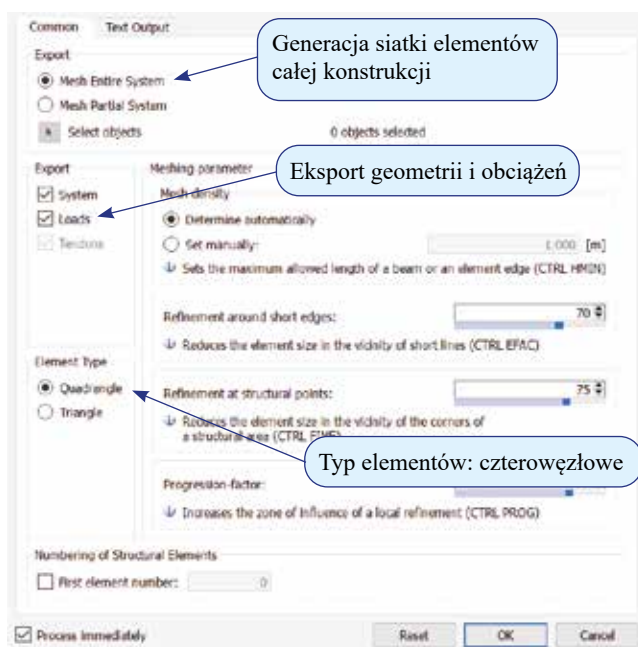
Ostatnim etapem tego przykładu jest wygenerowanie obciążenia ciężarem własnym. W tym celu należy przejść do zakładki *Obciążenia* (*Loads*) (rys. 5.23) i wybrać narzędzie *Menedżer przypadków obciążenia* (*Loadcase Manager*) (rys. 5.33). W pierwszej zakładce można zdefiniować naturę i kategorię obciążenia (*Actions*) (ciężar własny, eksploatacyjne, wiatr itp.), w drugiej tworzy się przypadki (*Loadcases*). Aby utworzyć przypadek obciążenia, należy kliknąć *Nowy* (*New*). Ciężar własny pojawia się domyślnie jako pierwszy przypadek; aby go aktywować, wystarczy wpisać wartość *Mnożnika ciężaru własnego* (*Factor of dead weight*) równą 1, ewentualnie zmienić również jego nazwę (jest to ważne zwłaszcza przy większej liczbie przypadków obciążenia i znacznie ułatwia późniejszą identyfikację). Wykorzystując powyższy tok postępowania, można stworzyć dowolną liczbę przypadków o określonej wcześniej naturze, a następnie za pomocą narzędzi do ich definicji (rys. 5.33) wygenerować odpowiednie schematy.



Rys. 5.33. Moduł SOFIPLUS-X – obciążenia



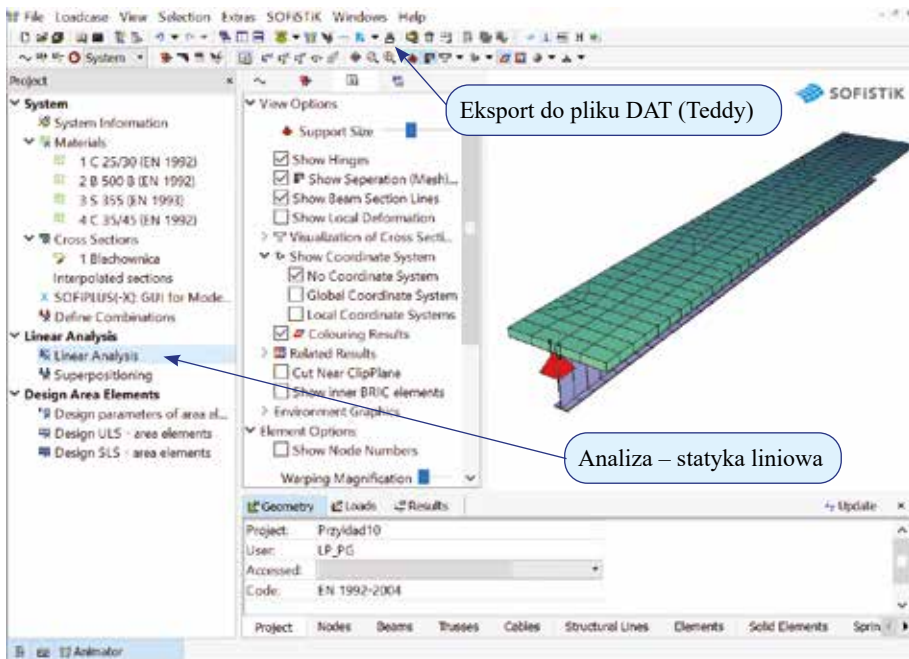
Rys. 5.34. Moduł SOFIPLUS-X – przypadki obciążeń



Rys. 5.35. Moduł SOFIPLUS-X – generator modelu

Wyczerpując wszystkie działania związane z definiowaniem geometrii i obciążeń, można przystąpić do generowania modelu obliczeniowego i eksportu danych do modułu *SSD*. W tym celu należy kliknąć ikonę *Generowanie modelu obliczeniowego, eksport danych do SSD* (*Create the calculation model by exporting the structure, generating loads, prestressing, etc.*) w górnej części drzewa wyboru (rys. 5.23). Następuje otwarcie okna generatora (rys. 5.35), w którym trzeba skontrolować ustawienia, a następnie zatwierdzić proces, klikając *OK*. Jeżeli wszystkie składowe modelu zostały wygenerowane poprawnie, można zapisać projekt stworzony w module *SOFIPLUS-X*, następnie go zamknąć i powrócić do modułu *SSD*.

Na rys. 5.36, korzystając z modułu *SSD*, przedstawiono wizualizację modelu obliczeniowego. Trzeba zauważyć, że mimo wyboru ustawienia „siatka regularna” uzyskano bardzo złej jakości siatkę nieregularną o znacznych obszarach dystorsji. Oczywiście, na ogół jest to związane ze złym doбором parametru wymiaru elementu, tu $h_{min} = 0,5$ m. Trzeba jednak podkreślić, że zależy to od wszechstronności przyjętego algorytmu oraz sposobu jego zakodowania i wbudowania w program.



Rys. 5.36. Moduł *SSD* – wizualizacja modelu (uwaga na kiepskiej jakości siatkę niestrukturalną)

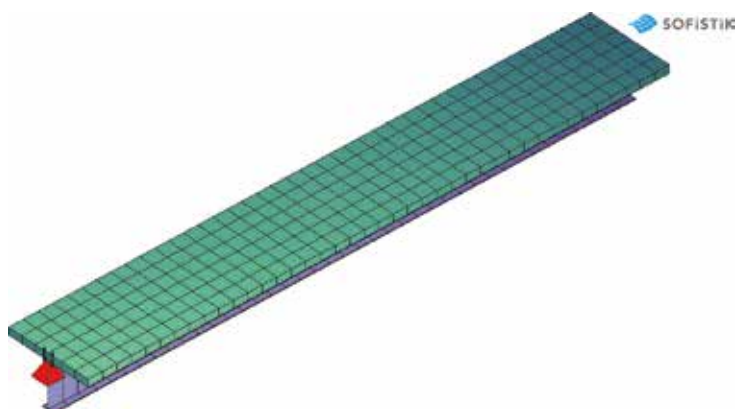
UWAGA. Regularność siatki, na ogół we wszystkich algorytmach, można jednak znacznie poprawić na innej drodze, mianowicie przez wprowadzenie większej liczby podobiektów geometrycznych (podobszarów) odwzorowujących się na tzw. prymitywy uwzględnione w programie. Ponadto, podobiektwy takie można wydzielić logicznie z układu, wykorzystując nabytą wiedzę o strukturalnej pracy konstrukcji. W uzupełnieniu trzeba dodać, że w takim postępowaniu w algorytmach musi być jeszcze przewidziane rozwiązanie problemu zapew-

nienia zgodności/ połączeń siatek podobszarów na ich granicach, szczególnie przy stosowaniu różnych gęstości podziału; niedopuszczalne jest pozostawianie tzw. węzłów sierocych.

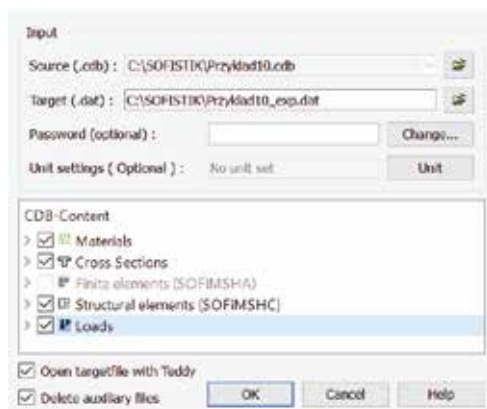
Generator wbudowany w module *SOFIPLUS-X* radzi sobie znacznie lepiej przy wprowadzeniu takich podobszarów. W analizowanym zadaniu logicznie konstrukcyjnie podziałem jest płyta przedzielona sztywnym żebrzem. A zatem pojedynczą powierzchnię płyty można zastąpić dwoma podobszarami rozgraniczonymi linią belki stalowej (rys. 5.37). W tym przypadku wystarcza to do uzyskania regularnej siatki podziału (rys. 5.38).



Rys. 5.37. Moduł *SOFIPLUS-X* – geometria płyty po podziale na dwa podobszary po linii żebra

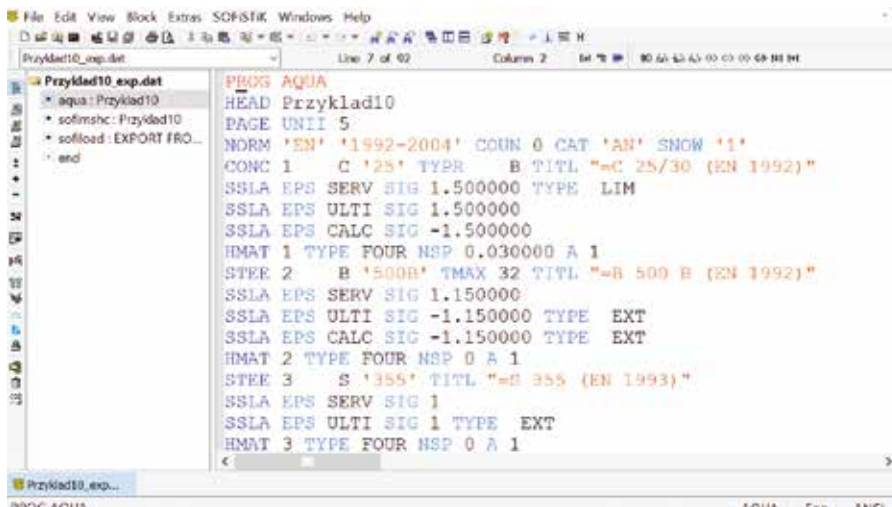


Rys. 5.38. Moduł *SSD* – wizualizacja geometrii po zmianie



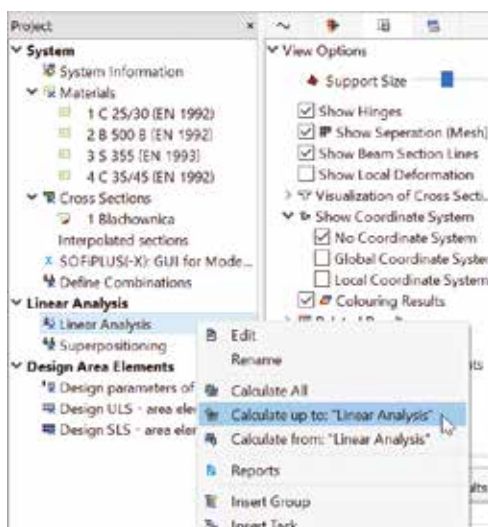
Rys. 5.39. Moduł *SSD* – eksport do pliku DAT

Pracę można kontynuować na dwa sposoby. Pierwszy sposób opiera się na już rozpoznany wcześniej podejściu. Polega on na wyeksportowaniu geometrii oraz obciążeń do pliku DAT (Export to DAT) (rys. 5.36, rys. 5.39) i dalszym rozwijaniu modelu obliczeniowego (np. obciążenia ruchome, obliczenia, kombinacje, obwiednie itd.) w module *Teddy* (rys. 5.40).

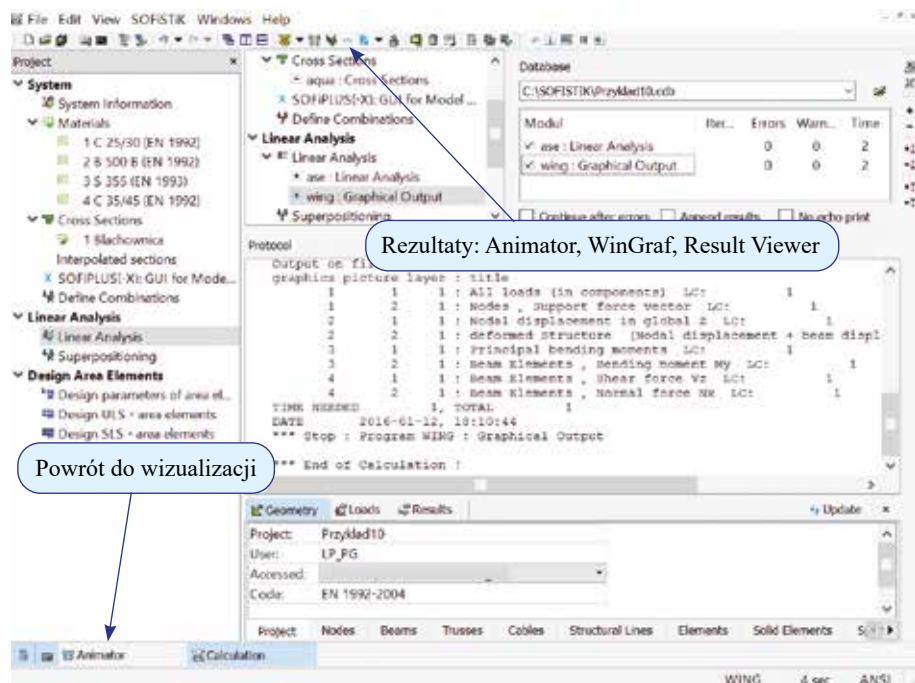


Rys. 5.40. Moduł *Teddy* – plik DAT z wyeksportowanym kodem

Drugi sposób to praca z poziomym modułem *SSD*. Kontynuację analizy uruchamia się poprzez kliknięcie prawym klawiszem na *Analiza – statyka liniowa (Linear Analysis)* (rys. 5.36), a następnie dokonuje się wyboru zakresu i przeprowadza obliczenia (rys. 5.41). Dostęp do wyników (moduły *Animator* oraz *WinGraf* oraz *Result Viewer*) jest możliwy bezpośrednio z okna *SSD* (rys. 5.42).



Rys. 5.41. Moduł *SSD* – wybór zakresu obliczeń



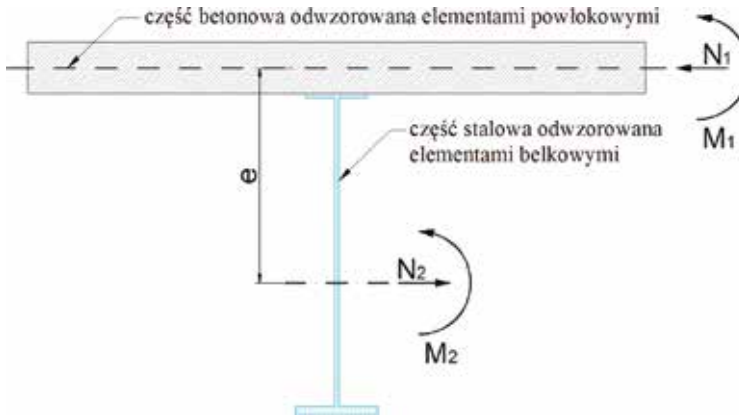
Rys. 5.42. Moduł SSD – obliczenia statyczne

5.4. Prezentacja wyników – siły wewnętrzne

W układach złożonych, tak jak w tym przykładzie, w którym blachownicę stalową opisują elementy belkowe, a płytę betonową – elementy powłokowe, siły wewnętrzne w całym przekroju zespolonym są udziałem dwóch typów elementów skończonych. Stąd, aby uzyskać siły wypadkowe w całym przekroju dźwigara zespolonego, niezbędne jest wykonanie dodatkowych obliczeń.

Przykładowo, aby otrzymać przekrojowy moment zginający, należy uwzględnić superpozycję udziału efektów z poszczególnych elementów, tj. momentów i sił normalnych na mimośrodku (zob. np. BIELEWICZ [2013]): $M = M_1 + M_2 + Ne$ (rys. 5.43). Jest to szczególnie przypadek czystego zginania, gdzie $N = N_1 = N_2$. Siły przekrojowe można również obliczyć z ich definicji na podstawie naprężeń lub zastosować wzór dla rozciągania mimośrodowego.

UWAGA. Siły przekrojowe można też uzyskać bezpośrednio, wykorzystując program SOFiSTiK. Należy zastosować moduł SIR. Przykładowy zapis przedstawiono poniżej. Kod ten umieszcza się w module *Teddy* po obliczeniach statycznych w programie ASE. Wynik można wyświetlić np. w module *WinGraf* jako *Results* → *External Beam Sections* → *Forces*.



Rys. 5.43. Belka zespolona – obliczanie sił przekrojowych

```
+prog sir
head M_przekrojowe
lc 1

sect no 1 xm 10 ym 0 zm 0 nx 1 ny 0 nz 0

end
```

Wywołanie programu *SIR*.

Numer przypadku obciążenia, dla którego ma być policzona siła przekrojowa.

Zadanie płaszczyzny: punkt i wektor normalny (rys. 5.44).

SECT			
Item	Description	Unit	Default
NO	Section number	-	1
XS	Beam ordinate	m	0.
XM	Global coordinates of the reference point	m	0.
YM		m	0.
ZM		m	0.
NX	Components of the section normal	-	0.
NY		-	0.
NZ		-	0.

Numer przekroju

Współrzędne globalne punktu do wstawienia przekroju

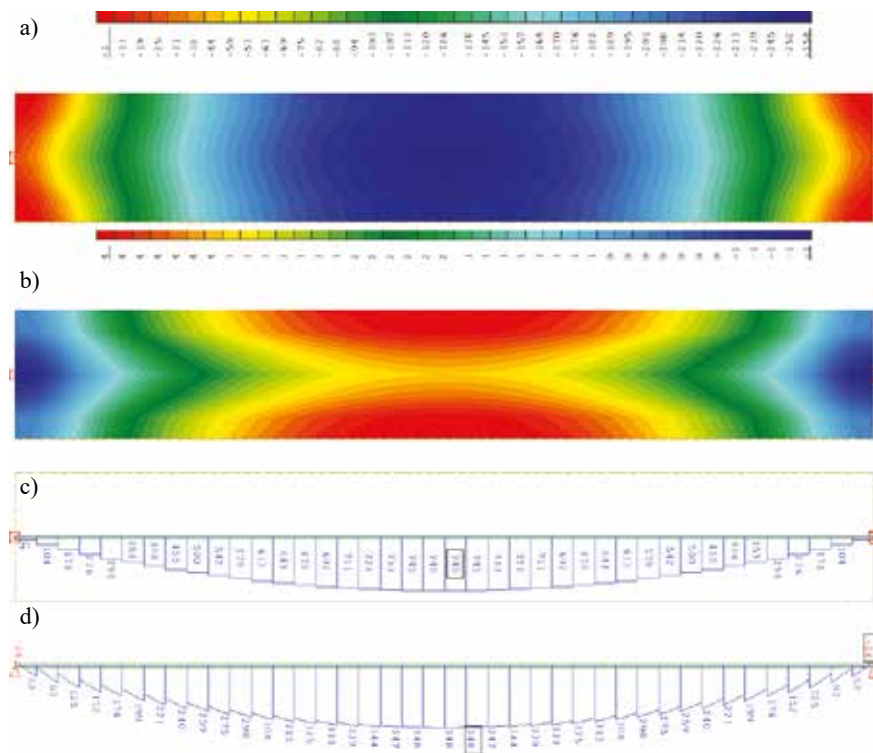
Składowe wektora normalnego do płaszczyzny przekroju

Rys. 5.44. Pomoc/ instrukcja do programu *SIR* – wybrane funkcje komendy *sect*

Prezentacja wyników w postaci wykresów lub map dla rozważanych układów jest analogiczna do zawartej w rozdziale 3.5 dla elementów prętowych oraz w rozdziale 4.3 dla elementów powłokowych. A zatem z wynikami można się zapoznać w module *Report Browser*, ewentualnie w modułach wynikowych *WinGraf* lub *Result Viewer* (rys. 5.42).

Na rys. 5.45 przedstawiono wyniki uzyskane z poziomu modułu *WinGraf*. Rezultaty odnoszące się do płyty betonowej (konwencja dla elementów powierzchniowych zob. rys. 4.18) mają formę map: składowej normalnej sił membranowych N_{xx} w kierunku lokalnej osi x zgodnej z osią belki zespolonej X (rys. 5.45a) i składowej momentu M_{xx} wywołującej

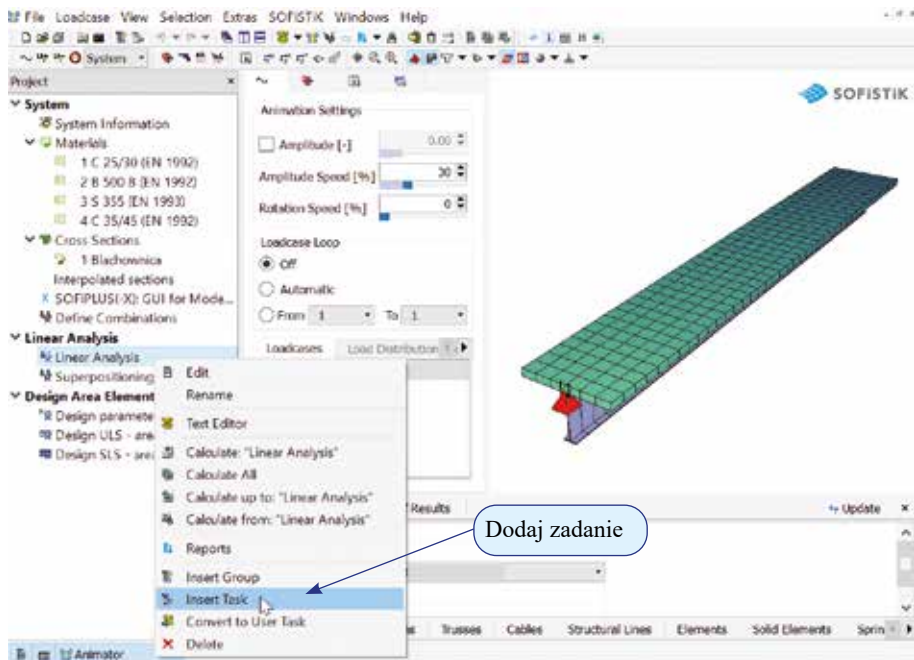
zginanie wzdłuż lokalnej osi x elementów powłokowych (rys. 5.45b). Wyniki odnoszące się do belki stalowej (konwencja dla elementów prętowych zob. rys. 3.32) przedstawiono w postaci wykresów: siły normalnej N (rys. 5.45c) i momentu zginającego M_y (rys. 5.45d) w elementach belkowych.



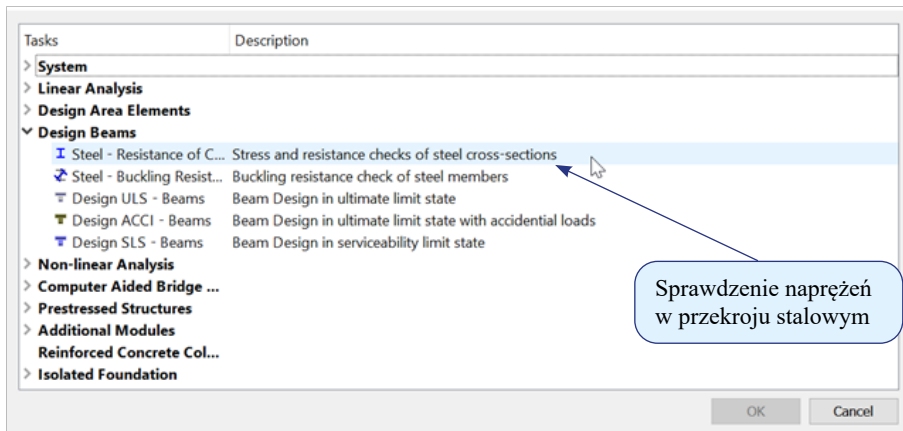
Rys. 5.45. Siły wewnętrzne – siły membranowe N_{xx} w płycie [kN/m] (a), momenty zginające M_{xx} w płycie [kNm/m] (b), siły normalne N w belce [kN] (c), momenty zginające M_y w belce [kNm] (d)

5.5. Prezentacja wyników – naprężenia



Sprawdzenia stanu naprężeń w przekrojach elementów prętowych także w tym wariancie zadania należy dokonać z użyciem programu *AQB*. W przypadku pracy w module *Teddy* wystarczy skopiować i uaktualnić kod danych z rozdziału 3.5. Natomiast jeśli analiza jest kontynuowana z poziomu modułu *SSD*, do drzewa zawierającego domyślne programy należy dołączyć nowe zadanie. W tym celu klika się prawym klawiszem myszy np. w *Analiza – statyka liniowa (Linear Analysis)*, a następnie wybiera opcję *Dodaj zadanie (Insert Task)* (rys. 5.46). Pojawia się okno wyboru (rys. 5.47), w którym należy zaznaczyć np. *Projektowanie konstrukcji stalowej (Design Steel Construction)*. Nowe zadanie zostaje wkomponowane w drzewo modelu (rys. 5.48). Ostatni krok to przeliczenie dodanego zadania, ewentualnie całego modelu (rys. 5.48).




Rys. 5.46. Moduł SSD – dodanie nowego zadania

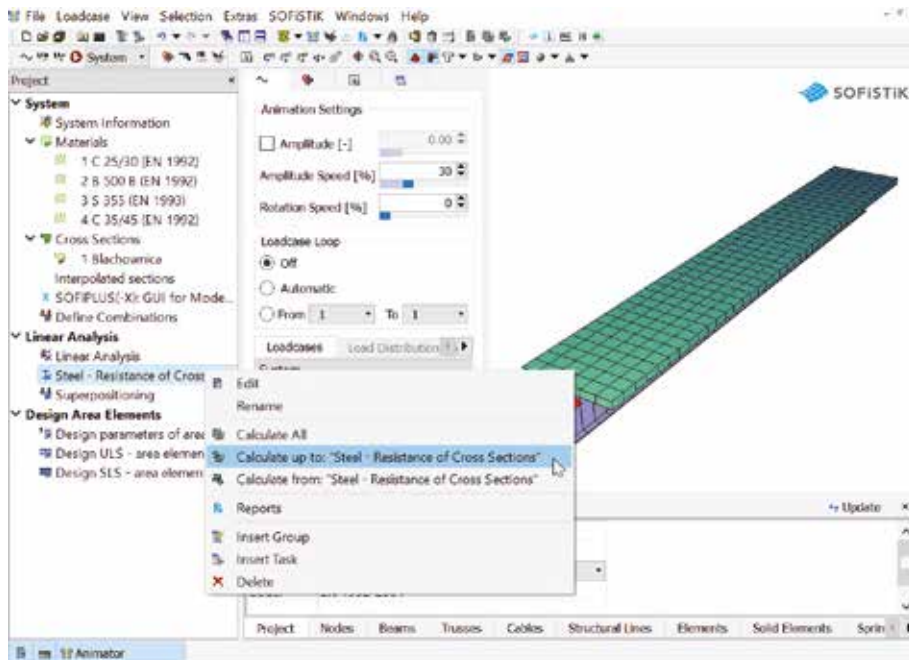


Rys. 5.47. Moduł SSD – wybór nowego zadania

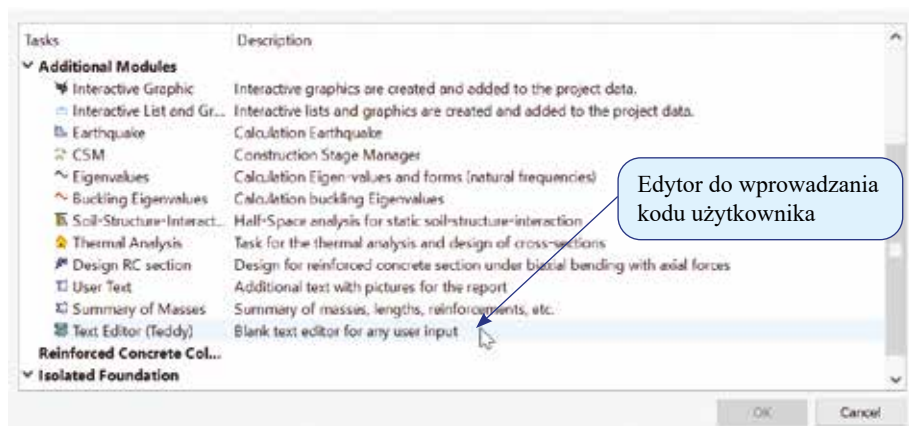
UWAGA. Zadanie to można również zrealizować z poziomu modułu SSD , wprowadzając do jego drzewka pusty dokument *Teddy*  (rys. 5.49, rys. 5.50). Po jego wkomponowaniu wystarczy w drzewku kliknąć dwa razy lewym klawiszem myszy w jego nazwę. W oknie obok otwiera się edytor tekstu, w którym można umieścić dowolny program, również *AQB* lub *MAXIMA*.

W celu uporządkowania zadania realizowanego w SSD  niewykorzystane narzędzia znajdujące się w drzewku: *Kombinacje+obwiednie* (*Superpositioning*) oraz całą grupę

Wymiarowanie elementów powierzchniowych (*Design Area Elements*) można usunąć, klikając w nie prawym klawiszem, a następnie wybierając opcję *Usuń (Delete)* (rys. 5.50).

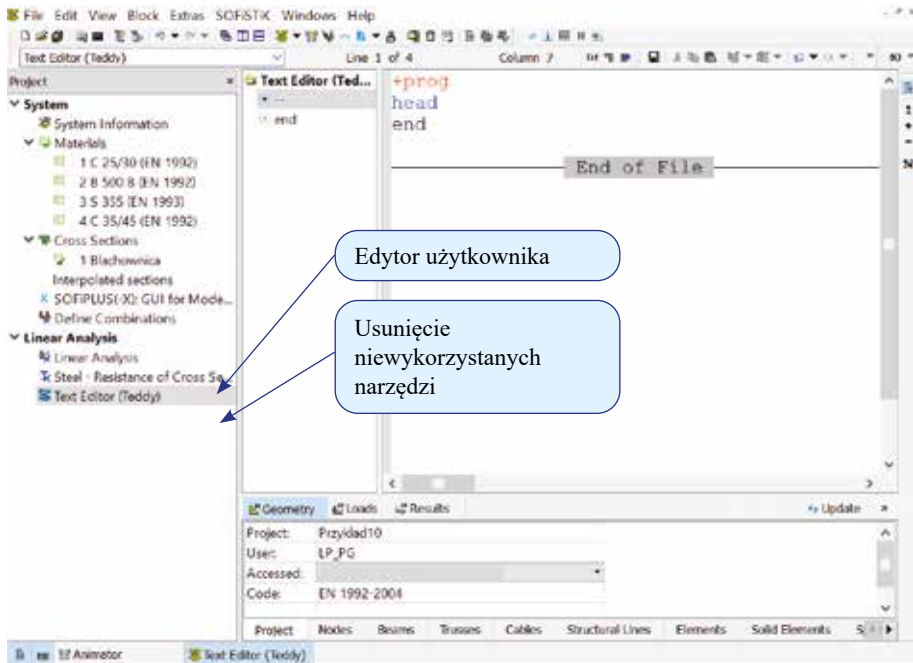
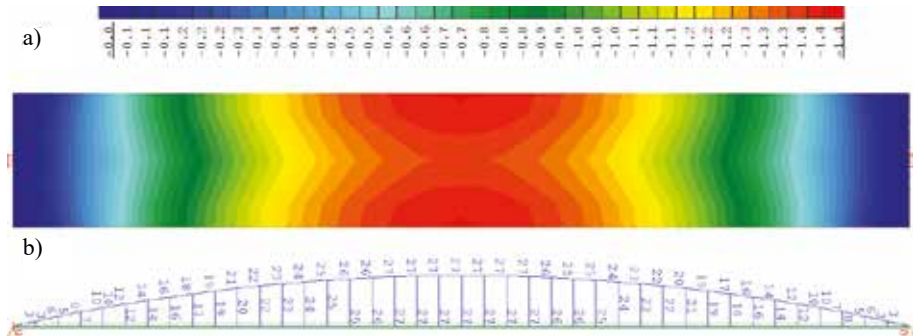


Rys. 5.48. Moduł SSD – obliczenia



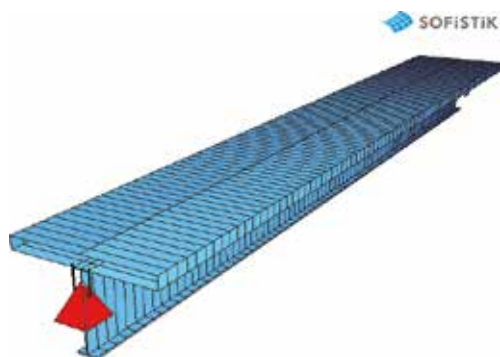
Rys. 5.49. Moduł SSD – dodanie pustego dokumentu *Teddy*

Na rys. 5.51 przedstawiono wynik w postaci mapy naprężeń składowej normalnej naprężeń σ_{xx} na kierunku lokalnej osi x na wierzchu płyty (rys. 5.51a) oraz wykres naprężeń normalnych na spodzie blachownicy (rys. 5.51b).

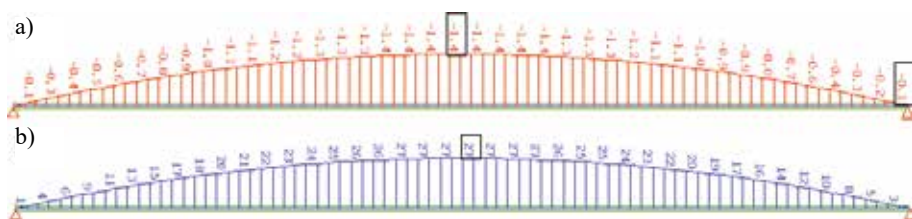
Rys. 5.50. Moduł SSD – edytor użytkownika *Teddy*Rys. 5.51. Naprężenia normalne σ_{xx} w kierunku lokalnej osi x [MPa]
– wierzch płyty (a), spód blachownicy (b)

Zadanie 5. Konstrukcja zespolona, układ prętowy

W ramach treningu stworzyć za pomocą dowolnej techniki kod do obliczeń dźwigara zespolonego o danych z przykładu 5.1 (rys. 5.1), wykorzystując wyłącznie elementy belkowe (zastosować przekrój zespolony, zob. rozdział 3.1, rys. 3.1e). Porównać wyniki sił wewnętrznych oraz stanu naprężenia z rezultatami uzyskanymi w modelu powłokowo-belkowym. Oczekiwany efekt przedstawiono w postaci wizualizacji w module *Animator* (rys. 5.52), przykładowy wynik zawarto na rys. 5.53.



Rys. 5.52. Moduł *Animator* – wizualizacja konstrukcji zespolonej opisanej tylko elementami belkowymi



Rys. 5.53. Naprężenia σ_{xx} w kierunku lokalnej osi x [MPa] – góra płyty (a), spód blachownicy (b)

6. ZADANIA DODATKOWE – ROZWIĄZANIA

Zadanie 1 – ruszt

```
+prog aqua
head przekroje+materialy
echo full
$ norma
norm en 1993-2005 00 unit 5
$ materialy
stee 1 s 355 $ stal
$ przekroje poprzeczne
sect 1 1 titl przekroj_a
poly opz
vert 1 0      0
      2 150   0
      3 150  20
      4 10    20
      5 10   1520
      6 200  1520
      7 200  1560
      8 0    1560

end
+prog sofimsha
head geometria
syst 3d
echo full
$ wezly definicja
node 1  0  0  0
      2  5  0  0
      3 10  0  0
      4  5 -5  0

$ elementy definicja
beam 101 1 2 1
      102 2 3 1
      103 2 4 1

$ podpory
node 1 - - - fix pp
      3 - - - fix xp
      4 - - - fix yp

end
```

Zadanie 2 – wspornik

```

+prog aqua
head przekroje+materialy
echo full
$ norma
norm en 1993-2005 00 unit 5
$ material
conc 1 c 45 titl c45/55 $ beton
end
+prog sofimsha
head geometria
syst 3d
echo full
$ wezly definicja
node 1 0 0 0
      2 0 1 0
      3 1 1 0
      4 1 0 0
      5 1 2 0
      6 2 2 0
      7 2 0 0
$ elementy definicja
quad mesh 1 2 3 4 1 posi cent t 250 m 5 n 5
quad mesh 4 5 6 7 1 posi cent t 250 m 10 n 5
$ podpory
boun 101 titl utwierdzenie
boun 1 2 fit f
end
+prog ase
head cw
lc 1 dlz 1 titl cw
end

```

Zadanie 3 – konstrukcja zespolona, SOFIMSHA

```

+prog aqua
head przekroje+materialy
echo full
norm en 1993-2005 00 unit 5 $ norma
$ material
ste 1 s 355 titl s355 $ stal
conc 2 c 35 titl c35/45 $ beton
$ przekroj poprzeczny
sect 1 mno 1 zm 250/2 titl przekroj_a
poly type opz
vert 1 0 0
      2 150 0

```

```

3      150   20
4      10    20
5      10   1520
6      200   1520
7      200   1560
8       0   1560

end
+prog sofimsha
head geometria
syst 3d
echo full
$ wezly definicja
let#nw 101
let#y -1.0
loop 10
  node (#nw #nw+40 1) (0 0.5) #y 0
  let#nw #nw+100
  let#y #y+0.5
endloop
$ elementy powlokowe definicja
let#nw 101
let#ne 1101
loop 9
  quad (#ne #ne+39 1) (#nw 1) (#nw+100 1) (#nw+101 1) (#nw+1 1)
mno 2 t 250 (przepisując kod do modułu Teddy, komendę quad zapisać w jednym wierszu!)
  let#nw #nw+100
  let#ne #ne+100
endloop
$ elementy pretowe definicja
beam (2101 2140 1) (301 1) (302 1) ncs 1 np -1
      (2201 2240 1) (801 1) (802 1) ncs 1 np -1
$ podpory
node 301 - - - fix pp
      341 - - - fix xp
      801 - - - fix pp
      841 - - - fix xp

end
+prog ase
head cw
lc 1 dlz 1 titl cw
end

```

Zadanie 4 – konstrukcja zespolona, SOFIMSHC

```

+prog aqua
head przekroje+materialy
echo full

```

```
$ norma
norm en 1993-2005 00 unit 5
$ material
stee 1 s 355 titl s355 $ stal
conc 2 c 35 titl c35/45 $ beton
$ przekroj poprzeczny
sect 1 mno 1 zm 0 titl przekroj_a
poly type opz
vert      1 0 0
          2 150 0
          3 150 20
          4 10 20
          5 10 1520
          6 200 1520
          7 200 1560
          8 0 1560

end
+prog sofimshc
head geometria
syst 3d
ctrl mesh 2+64; ctrl hmin 0.25 $0.5
$ punkty
spt 101 0 -1.0 0
    102 20 -1.0 0
    201 0 0 0 fix pp
    202 20 0 0 fix xp
    301 0 2.5 0 fix pp
    302 20 2.5 0 fix xp
    401 0 3.5 0
    402 20 3.5 0

$ linie wzdluzne
sln 1101 101 102
    1201 201 202 sno 1 eza 0.9192+0.25/2 eze 0.9192+0.25/2
    1301 301 302 sno 1 eza 0.9192+0.25/2 eze 0.9192+0.25/2
    1401 401 402

$ linie poprzeczne
sln 2101 101 401
    2102 102 402

$ powierzchnia
sar 1 mno 2 t 250 qref cent mctl 1
sarb out nl 1101,1401,2101,2102
end
+prog ase
head cw
lc 1 dlz 1 titl cw
end
```


Zadanie 5 – konstrukcja zespolona, porównanie modeli

```
+prog aqua
head przekroje+materialy
echo full
norm en 1993-2005 00 unit 5 $ norma
$ material
stee 1 s 355 titl s355 $ stal
conc 2 c 35 titl c35/45 $ beton
$ przekroj poprzeczny
sect 1 zm 0 titl przekroj_zespolony
$ plyta betonowa
poly type opz mno 2
vert 1 0 0
      2 1500 0
      3 1500 -250
      4 0 -250
$ blachownica
poly type opz mno 1
vert 1 0 0
      2 150 0
      3 150 20
      4 10 20
      5 10 1520
      6 200 1520
      7 200 1560
      8 0 1560
end
+prog sofimshc
head geometria
syst 3d
ctrl mesh 2+64; ctrl hmin 0.25
$ punkty
spt 101 0 0 0 fix ppmx
    102 20 0 0 fix xp
$ linie
sln 1101 101 102 sno 1
end
+prog ase
head cw
lc 1 dlz 1 titl cw
end
+prog aqb
head naprezenia
lc 1
stre e
end
```

7. BIBLIOGRAFIA

- BATHE K.J. (1996): *Finite Element Procedures*. Englewood Cliffs, Prentice-Hall.
- BIELEWICZ E. (2013): *Wytrzymałość materiałów*. Gdańsk, Wydawnictwo Politechniki Gdańskiej.
- CHRÓŚCIELEWSKI J., BURZYŃSKI S., DASZKIEWICZ K., SOBCZYK B., WITKOWSKI W. (2014): *Wprowadzenie do modelowania MES w programie ABAQUS. Materiały pomocnicze do laboratorium z metody elementów skończonych*. Gdańsk, Wydawnictwo Politechniki Gdańskiej.
- CHRÓŚCIELEWSKI J., MAKOWSKI J., PIETRASZKIEWICZ W. (2004): *Statyka i dynamika powłok wielopłatowych. Nieliniowa teoria i metoda elementów skończonych*. Biblioteka Mechaniki Stosowanej. Warszawa, IPPT PAN.
- CHRÓŚCIELEWSKI J., MIŚKIEWICZ M., PYRZOWSKI Ł (2010): *Metoda elementów skończonych – laboratorium*. Materiały dydaktyczne dla studentów umieszczone w serwisie OKNO Politechniki Gdańskiej.
- COOK R.D., MALKUS D.S., PLESHA M.E., WITT R.J. (2002): *Concepts and Applications of Finite Element Analysis*. New York, John Wiley & Sons 2002.
- HARTMANN F., KATZ C. (2002): *Statik mit finiten Elementen*. Berlin, Heidelberg, New York, Springer-Verlag.
- HARTMANN F., KATZ C. (2007): *Structural Analysis with Finite Elements*. Berlin, Heidelberg, New York, Springer-Verlag.
- HUGHES T.J.R. (2000): *The Finite Element Methods. Linear Static and Dynamic Finite Element Analysis*. Mineola, Dover Publications.
- MACNEAL R.H. (1978): A simple quadrilateral shell element. *Computers & Structures* 8, 175–183.
- MACNEAL R.H., WILSON C.T., HARDER R.L., HOFF C.C. (1998): The treatment of shell normals in finite element analysis. *Finite Elements in Analysis and Design* 30, 235–242.
- MIŚKIEWICZ M. (2016): *Nieliniowa analiza MES i monitoring konstrukcji prętowo-cięgnowych*. Gdańsk, Wydawnictwo Politechniki Gdańskiej.
- PYRZOWSKI Ł. (2011): *Viscoplastic damage analysis of structures subjected to impact loading. Plate and shell structures*. Saarbrücken, Lambert Academic Publishing.
- RACHOWICZ W. (2012): *Metoda elementów skończonych i brzegowych. Podstawy kontroli błędów i adaptacji*. Kraków, Wydawnictwo Politechniki Krakowskiej.
- RADWAŃSKA M. (2013): *Ustroje powierzchniowe. Podstawy teoretyczne oraz rozwiązania analityczne i numeryczne. Podręcznik dla studentów wyższych szkół technicznych*. Kraków, Wydawnictwo Politechniki Krakowskiej.
- RAKOWSKI G., KACPRZYK Z. (2005): *Metoda elementów skończonych w mechanice konstrukcji*. Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej.
- SOFiSTiK Manual, Version 2016-3, Software Version SOFiSTiK 2016.

-
- WILSON E.L. (2002): *Three-Dimensional Static and Dynamic Analysis of Structures. A Physical Approach with Emphasis on Earthquake Engineering*. Berkeley, Computers & Structures.
- ZIENKIEWICZ O.C. (1972): *Metoda elementów skończonych*. Warszawa, Arkady.
- ZIENKIEWICZ O.C., TAYLOR R.L. (2005): *The Finite Element Method*. 6th ed. Oxford, Elsevier.