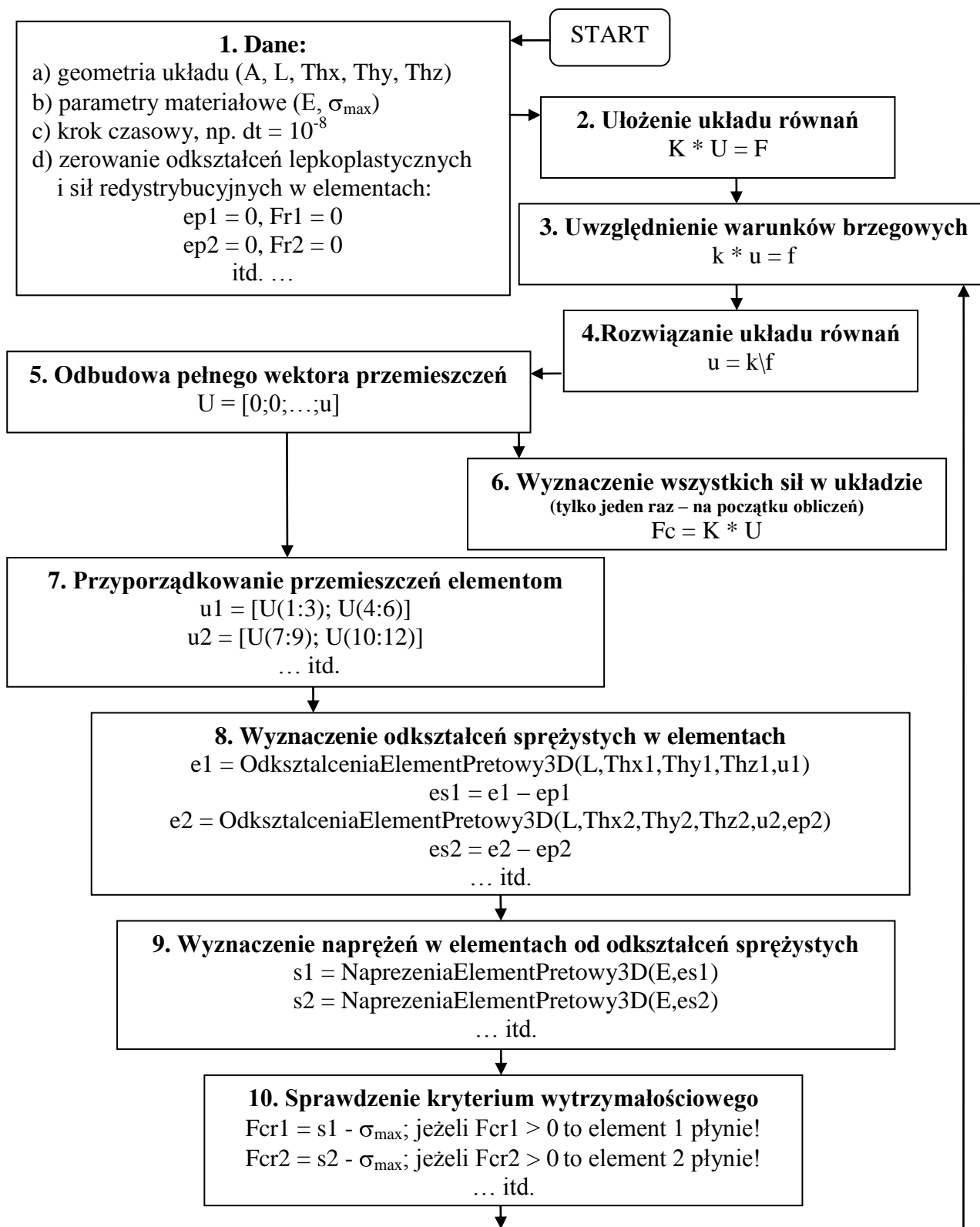


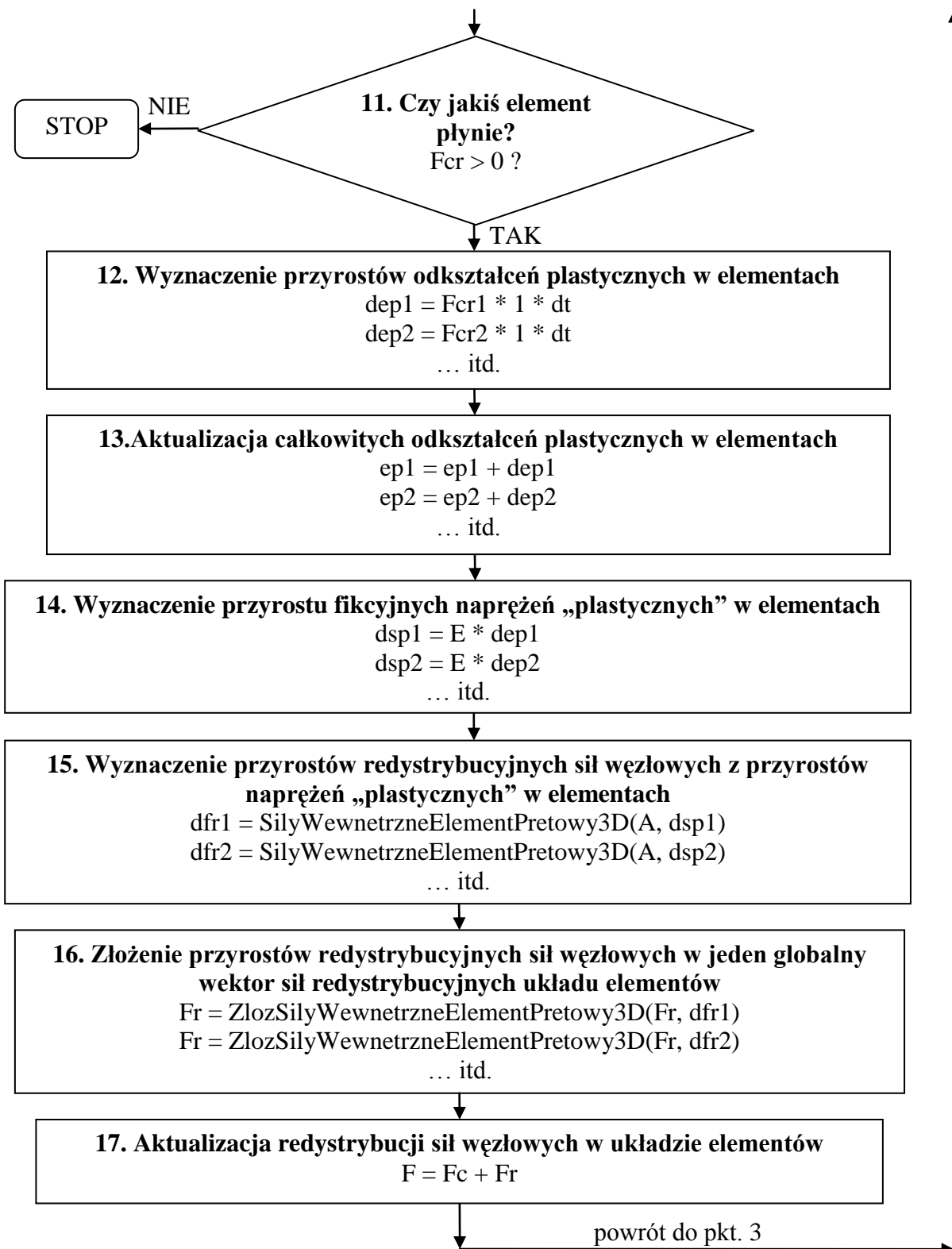
Metody Komputerowe

Metoda Elementów Skończonych

Element trójwymiarowy liniowy : pręt 3D – modelowanie sprężysto-plastyczności

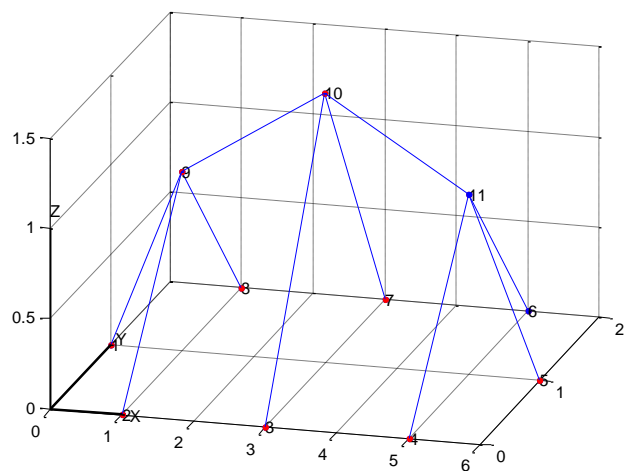
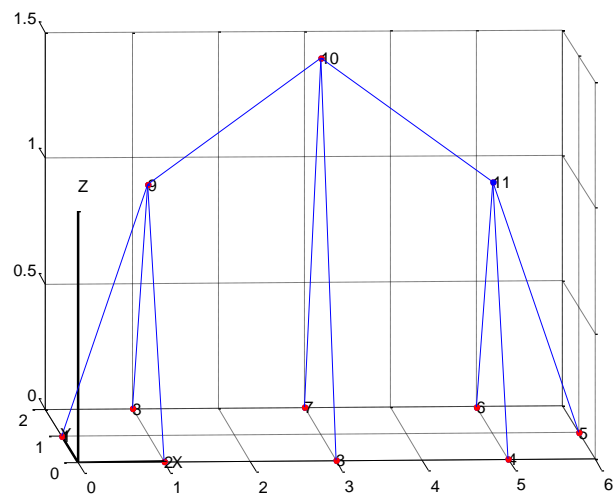
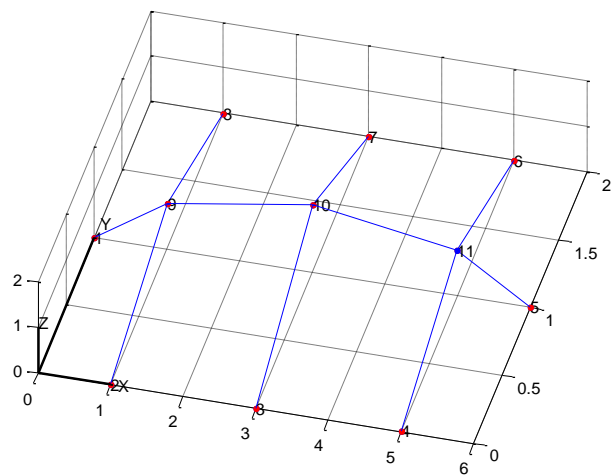
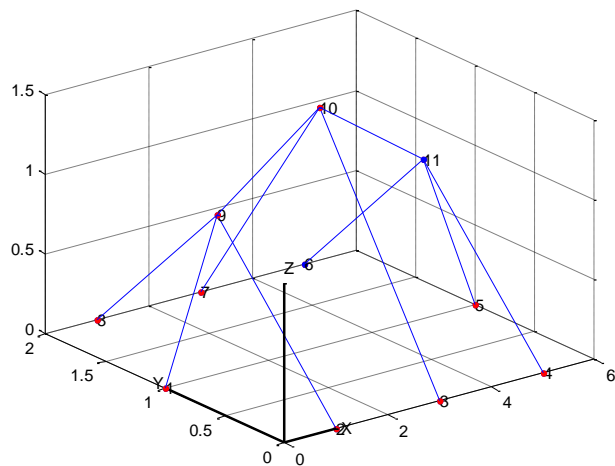
Najprostsze modelowanie plastycznego płynięcia materiałów sprowadza się do zastosowania iteracyjnej procedury odkształceń lepkoplastycznych, zwanej też procedurą odkształceń początkowych (ang. *initial strain*). Idea polega na ograniczaniu wartości naprężenia rozciągającego elementy do wartości wytrzymałości σ_{\max} poprzez wymuszenie plastycznego płynięcia przeciążonego elementu. Procedura ma następujący schemat:





Przykład nr 1.

Dla podanego układu elementów (rysunek) o polu przekroju $A = 19.2\text{cm}^2$ (kątowniki 100x100x10) wykonanych z materiału o znanym module $E = 210\text{GPa}$ i granicy plastyczności $S_{\text{max}} = 235\text{MPa}$ (stal S235JRG2), sprawdzić, czy możliwe jest przeniesienie obciążenia siłą $P_x = 700\text{kN}$ zaczepionego w węźle 10.



Rozwiązanie:

Krok 1 – dyskretyzacja zadania

Wszystkie polecenia przepisujemy do skryptu Matlaba, nadając mu nazwę „zadanie.m”. Skrypt powinien rozpoczynać się komendami czyszczącymi pamięć i okno komend:

```
clear;
clc;
```

po których wprowadzamy stałe materiałowe, geometryczne i obciążenie:

```
E = 210e6;
Smax = 235e3;
A = 19.2e-4;
Px = 700;
```

przyjmujemy mały krok czasowy:

```
dt = 8e-9;
```

oraz (zgodnie z algorytmem) zerujemy odkształcenia plastyczne elementów (w układzie mamy 10 elementów – można policzyć na rysunku):

```
ep = zeros(10,1);
```

a także wektor sił redystrybucyjnych (w układzie mamy 11 węzłów – czyli $11 * 3 = 33$ stopnie swobody):

```
Fr = zeros(33,1);
```

Następnie wprowadzamy współrzędne 11 węzłów (x,y,z):

```
x = [0 1 3 5 6 5 3 1 1 3 5];  
y = [1 0 0 0 1 2 2 2 1 1 1];  
z = [0 0 0 0 0 0 0 0 1 1.5 1];
```

Zadanie jest podzielone na 10 elementów następującą komendą:

```
elementy =[1 9; 2 9; 8 9; 9 10; 3 10; 7 10; 10 11; 4 11; 6 11; 5 11];
```

UWAGA! `elementy` to macierz, w której kolejne wiersze oznaczają kolejne elementy, a w każdym wierszu są numery dwóch węzłów definiujących dany element.

Poprawność wygenerowanej struktury można sprawdzić rysując układ elementów i węzłów:

```
plot3(x,y,z, '.')  
hold on;  
for k=1:11  
    info = sprintf('%d',k);  
    text(x(k),y(k),z(k),info);  
end  
showmesh3D(elementy,x,y,z,'r','b');  
pause  
close(gcf);
```

powinien pojawić się rysunek:

który zamkniemy naciskając dowolny klawisz.

Krok 2 – utworzenie macierzy sztywności dla każdego elementu i

Krok 3 – składanie macierzy sztywności elementów w jedną globalną macierz dla całego układu

W celu oszczędności miejsca w pamięci, generację danych geometrycznych elementów (długości L i kąty Thx , Thy , Thz), generację macierzy elementowych (k) i proces składania macierzy globalnej (K) przeprowadzimy w jednej pętli obliczeniowej:

```
K = zeros(33,33);  
  
for ie=1:10  
    w1 = elementy(ie,1);  
    w2 = elementy(ie,2);  
    %Ponizsze polecenie należy skopiowac W JEDNEJ LINII!  
    [L(ie),Thx(ie),Thy(ie),Thz(ie)] = DlugoscKatElementPretowy3D(x(w1),y(w1),  
        z(w1), x(w2), y(w2), z(w2));  
    k = SztynoscElementPretowy3D(E,A,L(ie),Thx(ie),Thy(ie),Thz(ie));  
    K = ZlozSztynoscPretow3D(K,k,w1,w2);  
end
```

Krok 4 – uwzględnienie warunków brzegowych i

Krok 5 – rozwiązywanie równań

Ponieważ podpory znajdują się w pierwszych ośmiu węzłach, to znamy przemieszczenia pierwszych $8*3=24$ stopni swobody (zero!). Zatem, wycinając te 24 pierwsze wiersze i kolumny z macierzy K , kopiujemy resztę do macierzy k (tzn. wiersze i kolumny od 25 do 33):

```
k = K(25:33, 25:33);
```

Analogicznie do K , przygotowujemy pusty wektor obciążeń F :

```
F = zeros(33,1);
```

który uzupełnimy wartością siły P_x działającą w kierunku x w 10 węzle, czyli w stopniu swobody nr $10 \cdot 3 - 2 = 28$:

$$F(28) = Px;$$

Analogicznie do K, wycinamy z wektora obciążeń F tylko te wiersze, które odpowiadają wyciętym wierszom i kolumnom macierzy k:

```
f = F(25:33);
```

Stworzony układ równań rozwiążemy poleceniem:

$$u = k \setminus f$$

wyznaczając wszystkie nieznane przemieszczenia.

Krok 6 – obróbka wyników (postprocessing)

Mając przemieszczenia wszystkich węzłów, możemy obliczyć reakcje w podporach. Najpierw zbierzmy przemieszczenia w jeden wektor (dodajemy do wyników zerowe przemieszczenia podpór):

[illegible]

a potem wyliczymy komplet sił w układzie:

$$F = K * U;$$
$$F_C = F;$$

który skopiujemy do \mathbb{F}_C i zachowamy „na później” (będziemy go „poprawiać” jeśli okaże się, że układ nie wytrzyma obciążenia P_x).

Teraz przygotujemy (wyzerujemy) wektor kryterium wytrzymałości, w którym będziemy przechowywać wartości przyjętej funkcji kryterium uplastycznienia (jedna liczba dla każdego z 10 elementów):

```
Fcr = zeros(10,1);
```

Wartości kryterium wytrzymałościowego wyznaczmy z naprężenia w każdym elemencie, które z kolei obliczymy na podstawie sprężystej części odkształceń wyznaczonych z kompletu przemieszczeń węzłów definiujących dany element (w jednej pętli obliczeniowej):

```
%dla każdego elementu od numeru 1 do 10
for ie = 1:10
    %numery wezlow elementu o numerze ie
    w1 = elementy(ie,1);
    w2 = elementy(ie,2);
    %numery stopni swobody
    dof1 = [w1.*3-2 w1.*3-1 w1.*3];
    dof2 = [w2.*3-2 w2.*3-1 w2.*3];
    %zebrane z U przemieszczenia elementu o numerze ie
    u = [U(dof1);U(dof2)];
endfor
```

```

%całkowite odkształcenie w elemencie o numerze ie
e(ie) = OdkształceniaElementPretowy3D(L(ie),Thx(ie),Thy(ie),Thz(ie),u);
%czesc sprężysta odkształcenia w elemencie o numerze ie
es(ie) = e(ie) - ep(ie);
%napreżenie od czesci sprężystej odkształcenia w elemencie o numerze ie
s(ie) = E * es(ie);
%funkcja kryterium wytrzymałościowego w elemencie o numerze ie
Fcr(ie) = s(ie) - Smax;
end

```

Sprawdzimy teraz, ile elementów ma przekroczone naprężenia S_{max} , tzn. w ilu elementach funkcja kryterium jest większa od zera:

```

nup = size(find(Fcr>0));
info = sprintf('Liczba uplastycznionych elementow = %d',nup(1));
disp(info);

```

i wyświetlimy maksymalną wartość funkcji kryterium w całym układzie elementów:

```

info = sprintf('Maksymalna wartość kryterium = %f',max(Fcr));
disp(info);

```

W układzie mamy jeden przeciążony element, a jego wartość kryterium (czyli „przeciążenie”) wynosi ok. 22799.347 kPa, tzn. 22.8 MPa – element uplastyczni się i „popłynie”. Ponieważ nie wiemy, który to element, użyjemy ogólnej procedury, która wyznaczy odkształcenia plastyczne oraz wynikającą z nich redystrybucję sił węzłowych w układzie dla każdego „przeciążonego” elementu:

```

%dla kazdego elementu od numeru 1 do 10
for ie=1:10
    %sprawdz, czy element o numerze ie jest przeciazany, tzn. czy Fcr > 0
    if Fcr(ie) > 0
        %element o numerze ie jest przeciazany
        %przyrost odkształcen plastycznych
        dep = Fcr(ie) .* 1 .* dt;
        %przyrost fikcyjnych naprezen od odkształcen plastycznych
        dsp = dep .* E;
        %aktualizacja całkowitych odkształcen plastycznych
        ep(ie) = ep(ie) + dep;
        %elementowe sily redystrybucyjne od przyrostu fikcyjnych naprezen
        fp = SilyWewnetrzneElementPretowy3D(A,Thx(ie),Thy(ie),Thz(ie),dsp);
        %złożenie elementowych sil redystrybucyjnych w jeden globalny wektor
        Fr = ZlozSilyWewnetrzneElementPretowy3D(Fr,fp,w1,w2);
    end
end
end

```

Mając globalny wektor sił redystrybucyjnych F_r , dokonujemy redystrybucji sił w układzie dodając F_r do uprzednio zachowanego wektora sił F_c (napisałem wcześniej, że się przyda ☺):

```

F = Fc + Fr;

```

i z tak powstałego wektora „poprawionych” sił (po redystrybucji „słabsze” elementy „oddały” część obciążenia „mocniejszym” elementom) wytniemy tylko ten fragment, który nie zawiera podpór (od 25 wiersza):

```

f = F(25:33);

```

Ponownie wyznaczymy przemieszczenia u w układzie, stosując jako obciążenie nowy wektor sił f (po redystrybucji obciążeń):

```
u = k\f;
```

Wyznaczone (nowe) u uzupełnimy o znane przemieszczenia w podporach:

[illegible]

i ponownie sprawdzimy wartości kryterium wytrzymałościowego F_{cr} w każdym elemencie, wyznaczając po kolei: z kompletu przemieszczeń $U \rightarrow$ odkształcenia sprężyste, z odkształceń sprężystych \rightarrow naprężenia, a z naprężeń \rightarrow funkcję kryterium plastyczności:

```
%dla każdego elementu od numeru 1 do 10
for ie = 1:10
    %numery węzłów elementu o numerze ie
    w1 = elementy(ie,1);
    w2 = elementy(ie,2);
    %numery stopni swobody
    dof1 = [w1.*3-2 w1.*3-1 w1.*3];
    dof2 = [w2.*3-2 w2.*3-1 w2.*3];
    %zebrane z U przemieszczenia elementu o numerze ie
    u = [U(dof1);U(dof2)];
    %całkowite odkształcenie w elemencie o numerze ie
    e(ie) = OdkształceniaElementPretowy3D(L(ie),Thx(ie),Thy(ie),Thz(ie),u);
    %część sprężysta odkształcenia w elemencie o numerze ie
    es(ie) = e(ie) - ep(ie);
    %napężenie od części sprężystej odkształcenia w elemencie o numerze ie
    s(ie) = E * es(ie);
    %funkcja kryterium wytrzymałościowego w elemencie o numerze ie
    Fcr(ie) = s(ie) - Smax;
end
```

Sprawdźmy teraz, ile elementów ma przekroczone naprężenia S_{max} , tzn. w ilu elementach funkcja kryterium jest większa od zera:

```
nup = size(find(Fcr>0));  
info = sprintf('Liczba uplastycznionych elementow = %d',nup(1));  
disp(info);
```

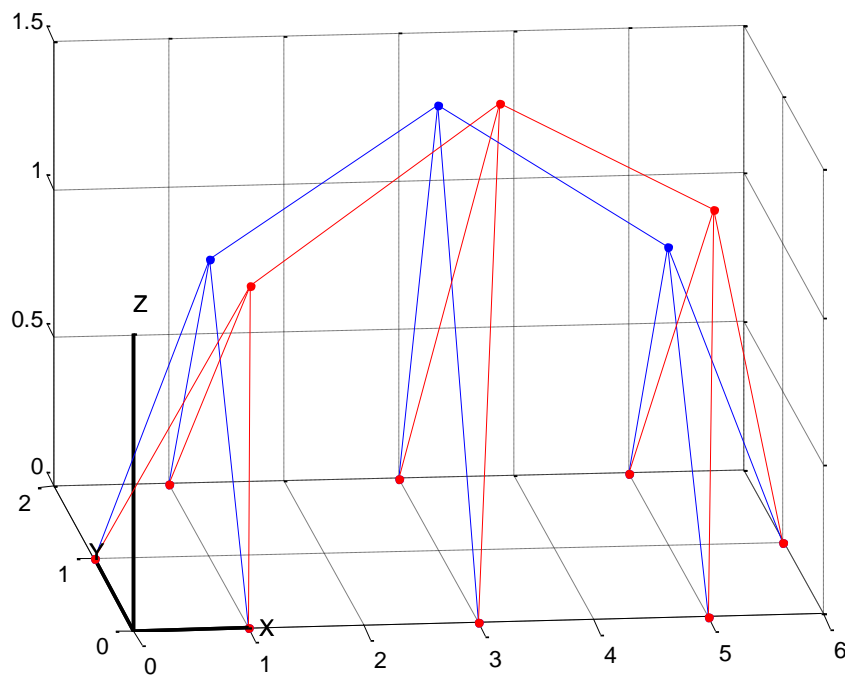
i wyświetlimy maksymalną wartość funkcji kryterium w całym układzie elementów:

```
info = sprintf('Maksymalna wartość kryterium = %f',max(Fcr));  
disp(info);
```

W układzie wciąż mamy jeden przeciążony element, ale jego wartość kryterium (czyli „przeciążenie”) wynosi tylko ok. 356.19 kPa, tzn. 0.36 MPa – chociaż element wciąż jest uplastyczniony i „płygnie”, to jednak wyraźnie spadły w nim naprężenia.

Odształcony układ możemy obejrzeć na wykresie stosując komendy:

```
showmesh3D(elementy,x,y,z,'g','g');
mnoznik = 20;
X = x + U(1:3:33)' .* mnoznik;
Y = y + U(2:3:33)' .* mnoznik;
Z = z + U(3:3:33)' .* mnoznik;
showmesh3D(elementy,X,Y,Z,'r','r');
pause
close(gcf);
```



UWAGA! Po naciśnięciu klawisza rysunek zostanie zamknięty.

Zadania do samodzielnego rozwiązania:

Zadanie nr 1.

Kontynuować obliczenia z Przykładu nr 1 do momentu, w którym wartość funkcji kryterium będzie mniejsza od 1. Ile iteracji obliczeniowych należało wykonać?

Zadanie nr 2.

Kontynuować obliczenia z Zadania nr 1 do momentu, w którym żaden z elementów nie będzie przeciążony. Ile iteracji obliczeniowych należało wykonać?

Zadanie nr 3.

Powtórzyć obliczenia z Przykładu nr 1 i Zadania nr 1 ale zakładając, że obciążenie układu $P_x = 900\text{kN}$. Co wynika z obliczeń (rysunek odkształconego układu)?