Metoda dystorsji wirtualnych, MES, algorytm Levenberga-Marquardta
optymalizacja ,konstrukcje kratownicowe
algorytm Gaussa-Newtona, metoda największego spadku

Roman KRÓL[1]

OPTYMALIZACJA KRATOWNIC PRZY UŻYCIU METODY DYSTORSJI WIRTUALNYCH I POPULARNYCH ALGORYTMÓW PRZEZNACZONYCH DO POSZUKIWANIA EKSTREMÓW FUNKCJI CELU

W artykule przedstawione zostały algorytmy optymalizacji funkcji wielu zmiennych oraz metoda dystorsji wirtualnych jako metoda szybkiej reanalizy (Fast Reanalysis Method) w odniesieniu do konstrukcji kratownicowych. Zaprezentowane zostały wyniki optymalizacji prostej kratownicy przeprowadzone w programie komputerowym przeznaczonym do trójwymiarowej analizy i optymalizacji kratownic. Artykuł zawiera porównanie efektywności algorytmów: Levenberga-Marquardta, Gaussa-Newtona oraz metody największego spadku. Zostały one zaimplementowane w programie GNU/Octave dla uzasadnienia wyboru algorytmu Levenberga-Marquardta wykorzystanego do poszukiwania konstrukcji najsztywniejszej przy ograniczonej ilości materiału oraz ograniczeniach na maksymalną absolutną wartość naprężeń.

OPTIMIZATION OF TRUSS STRUCTURES USING THE VIRTUAL DISTORTION METHOD AND POPULAR ALGORITHMS FOR SEARCH OF EXTREMA OF THE OBJECTIVE FUNCTION

This article presents optimization algorithms for multiple variables objective functions and the Virtual Distortion Method used in 3D truss structures. Simple three-element truss structure was optimized by computer program implemented for searching stiffest framework with volume and stress constraints. In this article efficiency of popular optimization algorithms was compared on the example of: Levenberg-Marquardt, Gauss-Newton and steepest descent method. Presented solutions was implemented in GNU/Octave to show the reason of using Levenberg-Marquardt algorithm for truss optimization with Virtual Distortion Method.

1. WSTĘP

Optymalizacja konstrukcji jest jedną z głównych dziedzin mechaniki wymagającą zastosowań techniki komputerowej. Opracowanie algorytmów dla rozwiązywania tego typu

[1] Politechnika Radomska im. Kazimierza Pułaskiego w Radomiu, Wydział Mechaniczny, ul. Krasickiego 54 26-600 Radom, e-mail: roman.krol@hotmail.com

zadań oprócz użycia klasycznych metod wymaga także zastosowania wspomagających technik obliczeniowych.

W komputerowej optymalizacji konstrukcji z użyciem metody elementów skończonych, istotnym problemem jest szybkość działania algorytmu. Potrzeba wielokrotnego uzyskania wyniku analizy statycznej przy modyfikacji któregoś z parametrów geometrycznych elementu wymaga częstego odwracania macierzy. Dla dużej liczby elementów skończonych operacja ta stanowi słaby punkt algorytmu. W celu rozwiązania tego problemu stosuje się metody szybkiej reanalizy (Fast Reanalysis Methods). Dzięki tym metodom, kosztem pewnej ilości pamięci można zdefiniować macierz pozwalającą na szybkie uzyskanie wyniku analizy bez potrzeby ponownego konstruowania macierzy sztywności.

Metody optymalizacji przedstawione w niniejszym artykule użyte zostały w programie komputerowym służącym do optymalizacji kratownic. Poszukuje on konstrukcji o najmniejszych przemieszczeniach węzłów przy ograniczeniach na maksymalną objętość oraz maksymalną absolutną wartość naprężeń w każdym z elementów.

2. METODA DYSTORSJI WIRTUALNYCH

W celu przyspieszenia optymalizacji w zadaniu przedstawionym w niniejszym artykule, użyta została metoda dystorsji wirtualnych [x]. Wymaga ona zbudowania przed procesem optymalizacji tzw. macierzy wpływu. Termin dystorsja dotyczy tu odkształcenia elementu kratownicy. Aby otrzymać macierz wpływu należy do każdego elementu wprowadzić dystorsje jednostkowe powodujące 100% odkształcenia elementów. Każda kolumna tej macierzy zawiera wektor odpowiedzi konstrukcji na dystorsję jednostkową wprowadzoną w określonym elemencie. Aby zastosować opisane zadanie optymalizacji należy wyrazić macierz wpływu w przemieszczeniach i odkształceniach.

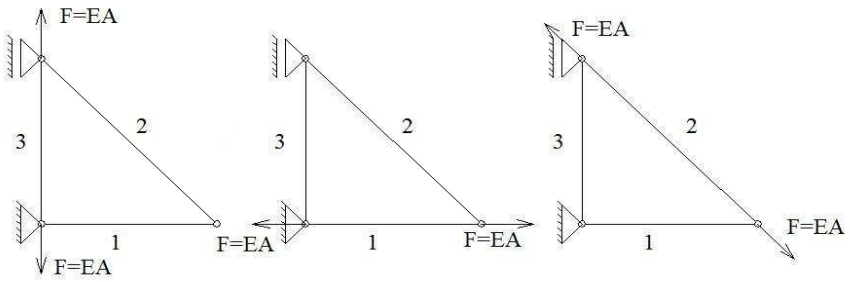
Duże przemieszczenia węzłów kratownicy znajdujące się w macierzy wpływu są podczas wyznaczania odpowiedzi konstrukcji skalowane przez wektor dystorsji $\varepsilon j0$ wyznaczany z równania $\mathbf{j\delta ij-1-\mu iDij\varepsilon j0=(1-\mu i)\varepsilon iL}$ (**1**).

$$\sum_j\left[\delta_{ij}-\left(1-\mu_i\right)D_{ij}\right]\varepsilon_j^0=(1-\mu_i)\varepsilon_i^L \tag{1}$$

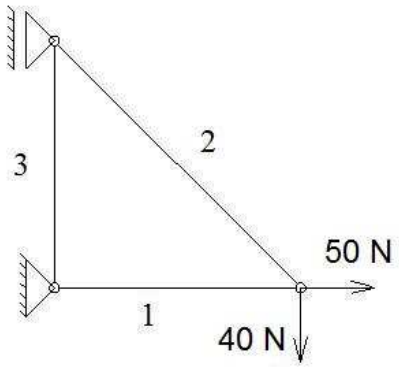
W powyższym równaniu D_{ij} oznacza element macierzy wpływu, μ_i to wektor argumentów funkcji celu opisanej równaniami (2) lub (3) (w tym przypadku wektor współczynników modyfikacji przekroju poprzecznego wyrażonych jako $\mu_i=\frac{A'}{A}$, gdzie A' oznacza pole przekroju poprzecznego zmodyfikowanego elementu, a A wyjściowe pole przekroju poprzecznego elementu), ε_i^L jest wektorem odpowiedzi konstrukcji obciążonej wyjściowymi siłami (przedstawionej na rysunku 2), wyrażonym w odkształceniach.

$$\varepsilon_i=\varepsilon_i^L+\sum_j D_{ij}\varepsilon_j^0 \tag{2}$$

$$u_i=u_i^L+\sum_j B_{ij}\varepsilon_j^0 \tag{3}$$



Rys.1. Sposób wprowadzania jednostkowych dystorsji w postaci pary sił rozciągających kolejno do elementów 1, 2, 3 kratownicy o parametrach z tabeli 1



Rys.2. Kratownica obciążona wyjściowymi siłami

Tab.1. Parametry kratownicy z rysunku 2

	Numer elementu		
	1	2	3
Długość elementu *l*[mm]	1000	1414	1000
Pole przekroju poprzecznego $A[mm^2]$	6400	6400	6400
Moduł Younga E[MPa]	$2 \cdot 10^5$	$2 \cdot 10^5$	$2 \cdot 10^5$

Tab.2. Przemieszczeniowa macierz wpływu dla kratownicy z rysunku 2

Przemieszczeniowe stopnie swobody	Wektory odpowiedzi [mm], po przeprowadzeniu analizy statycznej dla konstrukcji z dystorsją jednostkową wprowadzoną kolejno w każdym elemencie		
u_{1x}	0.0	0.0	0.0
u_{1y}	0.0	0.0	0.0
u_{1z}	0.0	0.0	0.0
u_{2x}	1000	-6.88e-14	0.0
u_{2y}	1000	-2000	1000
u_{2z}	0.0	0.0	0.0
u_{3x}	0.0	0.0	0.0
u_{3y}	0.0	0.0	1000
u_{3z}	0.0	0.0	0.0

Tab.3. Odkształceniowa macierz wpływu dla kratownicy z rysunku 2

Odkształcenia	Wektory odpowiedzi [mm], po przeprowadzeniu analizy statycznej dla konstrukcji z dystorsją jednostkową wprowadzoną kolejno w każdym elemencie		
ε_1	1.236	1.236	0.414
ε_2	0.414	1.236	0
ε_3	0	0	1

Bardziej szczegółowy opis metody dystorsji wirtualnych można znaleźć w [1-5]. Na podstawie zdefiniowanych powyżej równań (2) i (3) dla zadanego wektora argumentów μ_i można otrzymać sumę kwadratów przemieszczeń węzłów optymalizowanej konstrukcji.

3. ALGORYTMY OPTYMALIZACJI.

Korzystając z opisanej możliwości szybkiej reanalizy można zastosować klasyczne algorytmy optymalizacji funkcji wielu zmiennych. W programie komputerowym, w którym zastosowano opisane metody zaimplementowany został algorytm Levenberga-Marquardta [x] [x]. Dodatkowo wprowadzone zostały ograniczenia na maksymalną objętość oraz maksymalną absolutną wartość naprężeń w każdym elemencie konstrukcji za pomocą metody funkcji kary. Ograniczenia na nieujemne współczynniki modyfikacji przekroju poprzecznego wprowadzono korzystając z metody projekcji gradientu [x].

Oprócz algorytmu Levenberga-Marquardta często stosowane są także inne klasyczne algorytmy jak np. metoda największego spadku, algorytm Gaussa-Newtona czy algorytm L-BFGS. Zastosowany algorytm może być skutecznie stosowany dla funkcji o liczbie argumentów nie przewyższającej 1000 zmiennych.

Zaletą metody największego spadku (4) jest gwarantowana zbieżność. Jej użycie jest jednak w wielu przypadkach wolniejsze od innych przedstawionych w tym artykule algorytmów.

$$\boldsymbol{x_{i+1} = x_i - \lambda \nabla \Phi(x_i)} \tag{4}$$

Zaletą algorytmu Gaussa-Newtona (5) jest to, że nie wymaga on dostarczenia drugich pochodnych do obliczenia najmniejszej sumy kwadratów. Jego wadą w porównaniu do metody największego spadku jest jednak brak gwarancji zbieżności.

$$(\boldsymbol{J^T J})(\boldsymbol{x_{i+1}} - \boldsymbol{x_i}) = -\boldsymbol{\nabla f} \tag{5}$$

Algorytm Levenberga-Marquardta łączy w sobie zalety metody największego spadku i algorytmu Gaussa-Newtona. W nieliniowych obszarach optymalizowanej funkcji celu znaczący jest wpływ metody największego spadku. W przeciwnym przypadku rezultaty są zbliżone do wyników algorytmu Gaussa-Newtona. W algorytmie Levenberga-Marquardta wektor kierunku wyznaczany jest z równania (6), gdzie H oznacza przybliżony hesjan, f optymalizowaną funkcję a współczynnik λ zmienia się w kolejnych iteracjach decydując o rozmiarze wektora $x_{i+1} - x_i$.

$$\boldsymbol{x_{i+1}} = \boldsymbol{x_i} - \{\boldsymbol{H(x_i)} + \boldsymbol{\lambda diag[H]}\}^{-1}\boldsymbol{\nabla f}(\boldsymbol{x_i}) \tag{4}$$

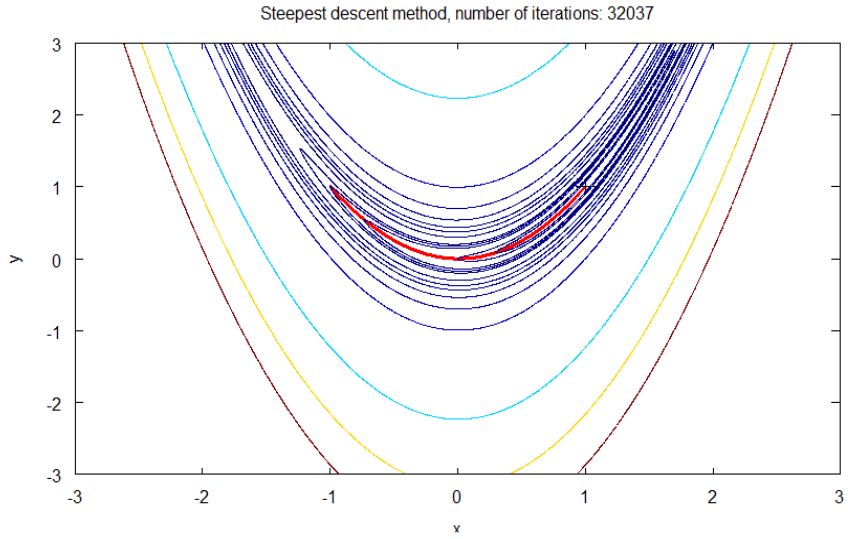
Te popularne algorytmy optymalizacji często bywają używane wraz z dodatkowymi metodami, takimi jak liniowe przeszukiwanie oraz metoda projekcji gradientu. Opisane w tym punkcie algorytmy zaimplementowane zostały także w postaci skryptów dla programu GNU/Octave w celu przeprowadzenia porównania efektywności. W dalszej części artykułu przedstawione zostanie porównanie wyników zastosowania metody największego spadku, algorytmu Gaussa-Newtona i algorytmu Levenberga-Marquardta na przykładzie funkcji Rosenbrocka (7).

$$f(x, y) = (1 - x)^2 + 100 \cdot (y - x^2)^2 \tag{7}$$

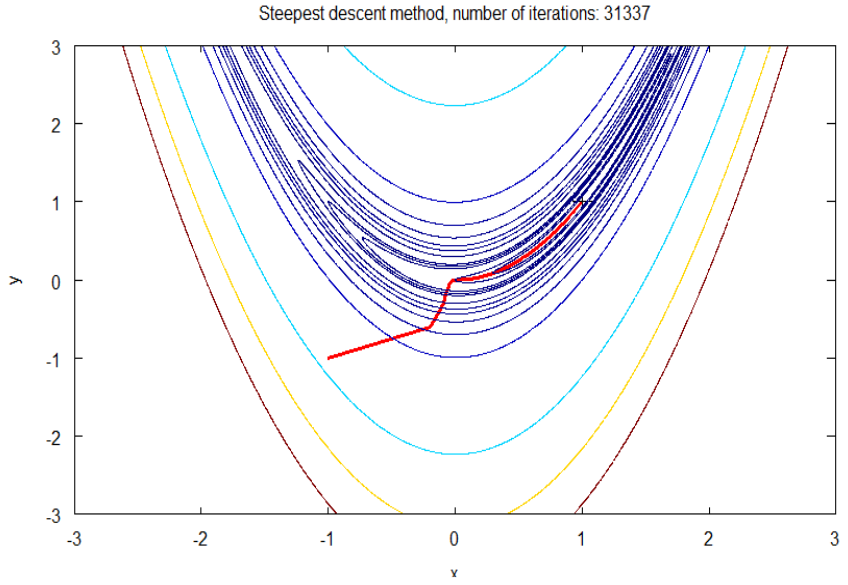
Wyniki przedstawione zostały na rysunkach 3-7 i w tabeli 4.

Tab. 4. Porównawcze wyniki efektywności metody największego spadku oraz algorytmów: Gaussa-Newtona i Levenberga-Marquardta

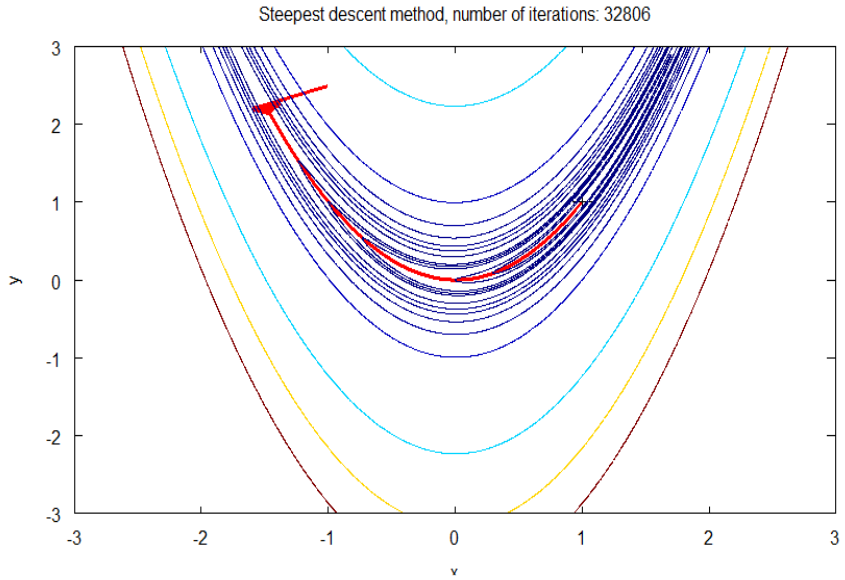
Punkt startowy	metoda największego spadku	algorytm Gaussa-Newtona	algorytm Levenberga-Marquardta
	liczba iteracji		
(-1,1)	32037	17438	2995
(-1,-1)	31337	81549	18050
(1,-1)	31284	118950	25851
(-1,2.5)	32806	61224	17592



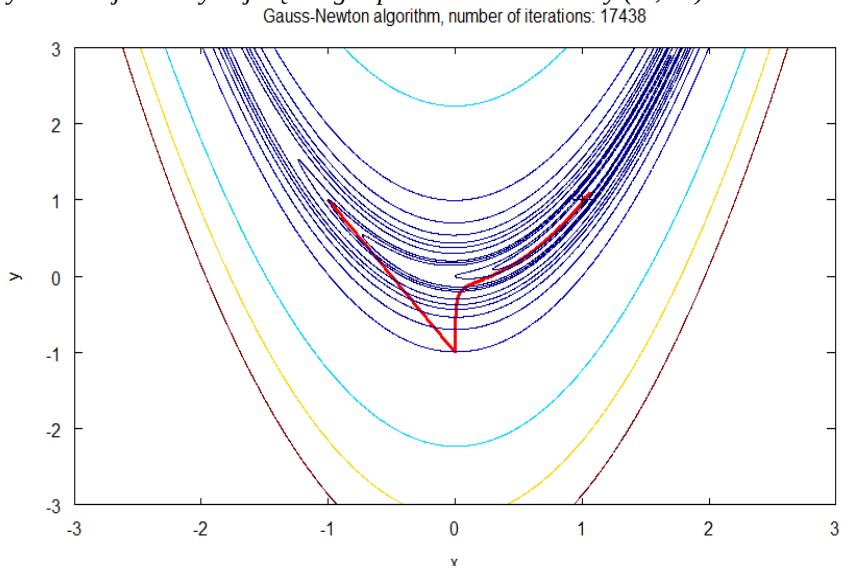
Rys. 3. Wykres warstwicowy funkcji Rosenbrocka (7). Czerwona linia łączy punkty kolejnych iteracji metody największego spadku. Punkt startowy (-1,1)



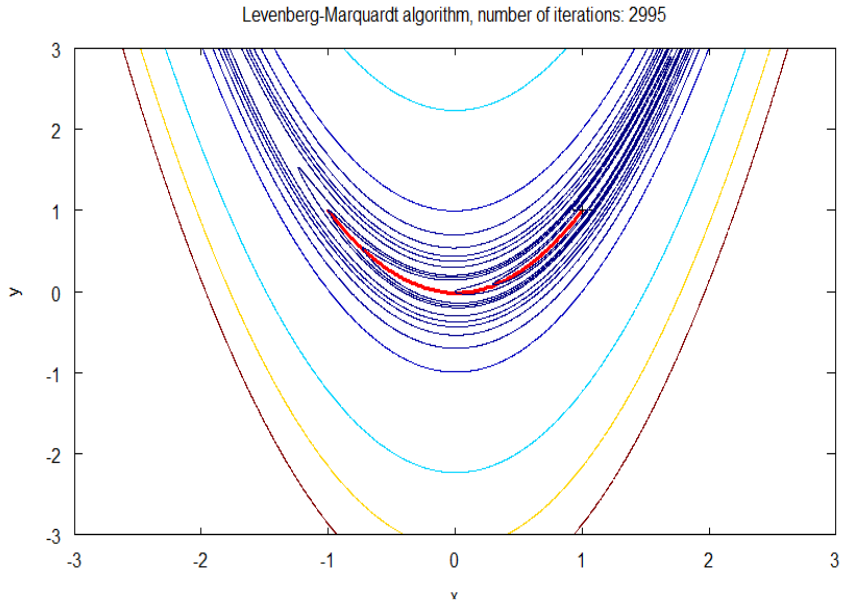
Rys. 4. Wykres warstwicowy funkcji Rosenbrocka (7). Czerwona linia łączy punkty kolejnych iteracji metody największego spadku. Punkt startowy (-1,-1)



Rys. 5. Wykres warstwicowy funkcji Rosenbrocka (7). Czerwona linia łączy punkty kolejnych iteracji metody największego spadku. Punkt startowy (-1,2.5)



Rys. 6. Wykres warstwicowy funkcji Rosenbrocka (7). Czerwona linia łączy punkty kolejnych iteracji algorytmu Gaussa-Newtona. Punkt startowy (-1,1)



Rys. 7. Wykres warstwicowy funkcji Rosenbrocka (7). Czerwona linia łączy punkty kolejnych iteracji algorytmu Levenberga-Marquardta. Punkt startowy (-1,1)

Z rysunków 3-7 oraz na podstawie wyników przedstawionych w tabeli 4 można wywnioskować, że metoda największego spadku jest wolniejsza od pozostałych algorytmów, ale gwarantuje zbieżność. Algorytm Gaussa-Newtona dla punktu startowego (-1,1) okazał się szybszy od metody największego spadku, ale jego efektywność dla punktów startowych znajdujących się w nieliniowych obszarach funkcji celu zależy od odpowiedniego wyboru współczynnika skalującego długość kroku. Współczynnik ten powinien być odpowiednio mały. Algorytm Levenberga-Marquardta okazał się w przedstawionym przypadku najefektywniejszy. W praktyce testowania tego algorytmu dla kratownic zdarzały się przypadki znajdowania lokalnych minimów przez wpływ metody największego spadku w określonych warunkach. Algorytm ten nie gwarantuje także zbieżności dla wektora wstępnego oszacowania wyniku odległego od poszukiwanego globalnego minimum.

W tabeli 5 przedstawiono wyniki optymalizacji kratownicy z rysunku 2, o parametrach zawartych w tabeli 1.

Tab. 5. Wyniki optymalizacji kratownicy o parametrach przedstawionych w tab. 1.

Numer elementu	współczynniki modyfikacji pola przekroju poprzecznego μ_i kolejnych elementów
1	0.69
2	1.6
3	1.85

4. PODSUMOWANIE

Z testowania efektywności trzech algorytmów optymalizacji: Levenberga-Marquardta, metody największego spadku i Gaussa-Newtona wynika, że pierwszy z wymienionych jest najszybszy. Liczba iteracji w algorytmie Levenberga-Marquardta jest kilkakrotnie mniejsza w porównaniu z algorytmami metody największego spadku i Gaussa-Newtona. Metoda największego spadku gwarantuje zbieżność lecz jest zbyt wolna do obliczeń konstrukcji.

Algorytm Levenberga-Marquardta łączy w sobie cechy metody największego spadku i algorytmu Gaussa-Newtona. Jego wadą jest znajdowanie minimów lokalnych w sytuacji, kiedy zaznacza się wpływ metody największego spadku. Istotny dla tego algorytmu jest także odpowiedni wybór wektora wstępnego oszacowania wyniku. Jeśli jest on odległy od optymalnego rozwiązania, to może negatywnie wpłynąć na zbieżność.

Optymalne rozwiązanie w tabeli 5 wskazuje na to, że przy obciążeniu kratownicy jak na rysunku 2 najistotniejsze dla sztywności konstrukcji są pręty: pionowy oraz przeciwprostokątny. Dla przedstawionego rozwiązania wykorzystana została całkowita dostępna objętość materiału V=30700000 mm^3 (dokładność ustalana była za pomocą współczynników funkcji kary).

5. BIBLIOGRAFIA

[1] Holnicki-Szulc, J. & Gierliński, J. *Structural Analysis, Design and Control by the Virtual Distortion Method, John Wiley & Sons Ltd*, Chichester, 1995.

[2] Holnicki-Szulc, J. (ed.) *Smart Technologies for Safety Engineering*, John Wiley & Sons Ltd, Chichester, 2008.

[3] Kołakowski, P.; Wikło, M. & Holnicki-Szulc, J. *The virtual distortion method – a versatile reanalysis tool for structures and systems. Structural and Multidisciplinary Optimization*, 2008, 36(3), 217-234.

[4] Akgün, M. A.; Garcelon, J. H. & Haftka, R. T. *Fast exact linear and non-linear structural reanalysis and the Sherman-Morrison-Woodbury formulas, International Journal for Numerical Methods in Engineering*, 2001, 50(7), 1587-1606.

[5] Zieliński, T. G. *Metoda Impulsowych Dystorsji Wirtualnych z zastosowaniem do modelowania i identyfikacji defektów w konstrukcjach*, Instytut Podstawowych Problemów Techniki PAN, Warszawa, 2003.

[6] Kołakowski P. *Analiza wrażliwości i optymalne projektowanie konstrukcji kratowych metoda dystorsji wirtualnych*, Instytut Podstawowych Problemów Techniki PAN, Warszawa, 1998.

[7] Manolis I. A. Lourakis; Antonis A. Argyros *The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm*, www.ics.forth.gr/lourakis/sba, FORTH-ICS/TR-340, 2004.

[8] Manolis I. A. Lourakis *Levenberg-Marquardt nonlinear least squares algorithms in C/C++*, http://www.ics.forth.gr/~lourakis/levmar/, 2004-2010.