

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Algorytmy Ewolucyjne

Projekt 1

Sofiya Makarenka

Spis treści

Treść zadania.	3
Krótki opis wybranych metod.	3
Analityczne rozwiązanie problemu.	4
Funkcja fminsearch - metoda Nelder-Mead.	5
Funkcja fminunc - metoda obszaru zaufania - analitycznie obliczona macierz Hessego i gradient.	7
Funkcja fminunc - metoda quasi-Newton - estymowany przez algorytm gradient.	9
Funkcja fminunc - metoda quasi-Newton - estymowany przez algorytm gradient.	11
Ocena działania metod.	13
References.	14

1. Treść zadania.

Znajdź minimum funkcji Rosenbrock'a („bananowej”) bez ograniczeń:

$$f(x) = (1 - x + a)^2 + 100 [y - b - (x - a)^2]^2$$

Porównaj działanie dostępnych (w Matlabie) co najmniej czterech metod optymalizacji gładkiej bez ograniczeń

Wyniki przedstaw w postaci sprawozdania (pdf) z wynikami obliczeń

2. Dane.

“Projekt należy wykonać w parach (...). W przypadku wykonania projektu w parze należy wybrać ten numer wiersza z tabeli, który odpowiada niższemu numerowi indeksu.”

Niższy numer indeksu w naszej parze - 304135.

Odpowiadające mu dane:

Nr.	a	b	X1	Y1	X2	Y2	X3	Y3	X4	Y4
8	-1	-1	1	0	0	-2	-2	-2	-2	0

3. Krótki opis wybranych metod.

- **metoda Neldera-Meada**

Jest to metoda tak zwanego poruszającego sympleksu. Sympleks składa się z $N+1$ wierzchołków, gdzie $N - \mathbb{R}^n$, czyli w naszym wypadku $N = 3$. Metoda na początku tworzy sympleks wokół punktu startowego, a dalej modyfikuje ten sympleks zgodnie z algorytmem opisanym w dokumentacji matlabowej. Jest to prosta i skuteczna dla małych obszarów metoda, bez dowodu zbieżności. Ta metoda nie wykorzystuje gradientu.

- **metoda quasi-Newtona z estymowanym przez algorytm**

Jest to metoda gradientowa. Matlab sam estymuje gradient i macierz Hessego, więc nie musimy wprowadzać tego ręcznie. Hesjan jest przybliżany przez analizowanie kolejnych wektorów gradientu. Algorytm szuka punktu, w którym gradient funkcji celu jest równy zeru.

- **metoda zaufanego obszaru - wprowadzenie gradientu i hessianu**

Algorytm aproksymuje funkcję celu prostszą funkcją, która odzwierciedla zachowanie funkcji celu wokół punktu x - to jest tak zwany obszar zaufania. Aproksymacja funkcją zwykle przebiega za pomocą szeregu Taylora, gradientu i macierzy Hessego.

4. Analityczne rozwiązanie problemu.

Gradient i macierz Hessego obliczyliśmy za pomocą napisanego przez nas pliku matlabowego: gradientHessianFinder.m. Wyniki są następujące

$$\begin{aligned}\nabla f &= [2x - 2a - 2 - 400(a - x)(b - y + (a - x)^2) ; \\ &\quad 200(y - b - (a - x)^2)]; \\ H(x,y) &= [400(3(x - a)^2 - y + b) + 2, 400(a - x); \\ &\quad 400(a - x), \quad 200];\end{aligned}$$

Jeżeli chodzi o analitycznym rozwiązaniu funkcji celu, to skoro mamy sumę dwóch kwadratów, to minimalna wartość funkcji wynosi 0, możemy tego osiągnąć przyrównując obydwa składniki do zera.

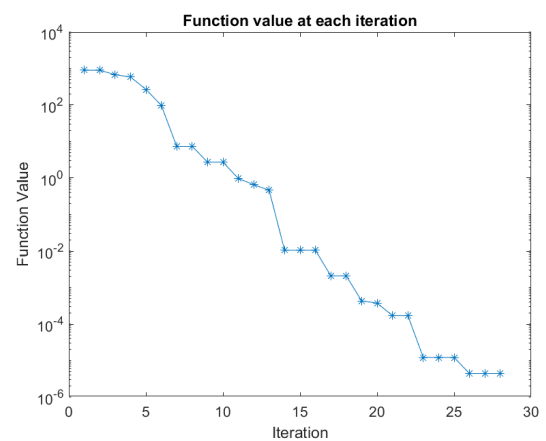
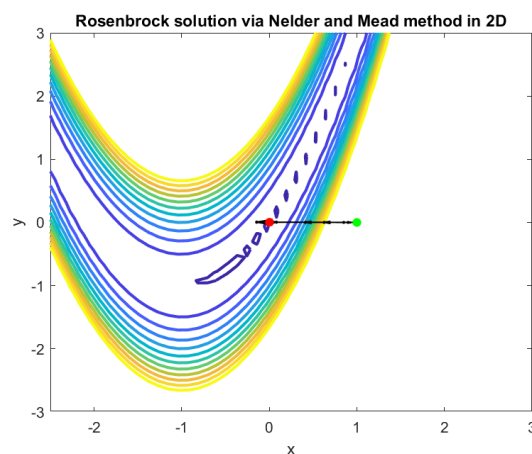
$$\begin{aligned}(1 - x + a)^2 &= 0 \\ [y - b - (x - a)^2]^2 &= 0\end{aligned}$$

Rozwiązaniem tego układu równań jest punkt - (0,0).
Obliczając dokładności, używałyśmy błędu średniokwadratowego.

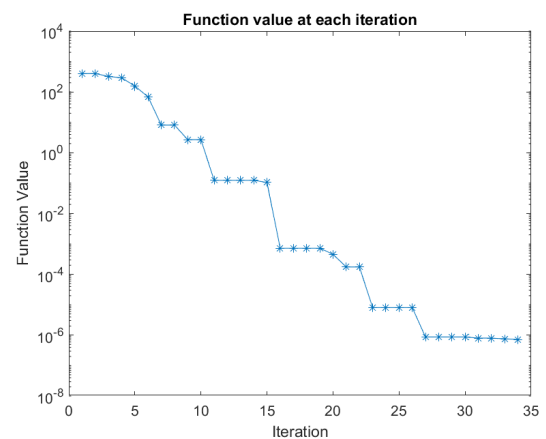
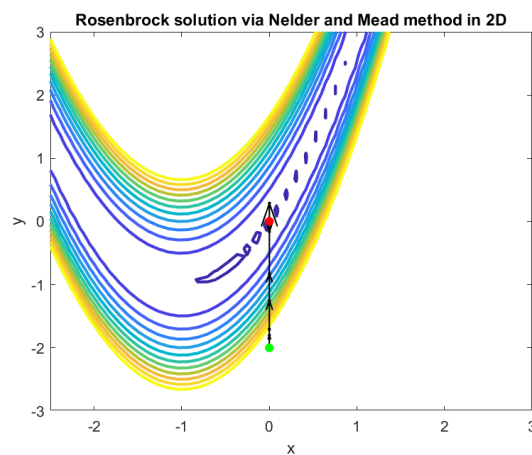
5. Funkcja fminsearch - metoda Nelder-Meada.

Punkt Startowy	Punkt końcowy	Liczba iteracji	Dokładność
(1,0)	(0.0020, 0.0040)	27	2.0342e-05
(0,-2)	(8.5255e-04, 17.00e-04)	33	3.6403e-06
(-2,-2)	(2.1988e-06, 5.5380e-04)	75	3.5504e-11
(-2,0)	(2.5561e-05, 4.9382e-05)	104	3.0920e-09

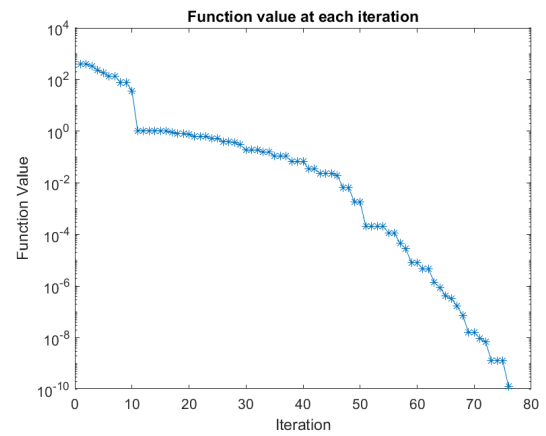
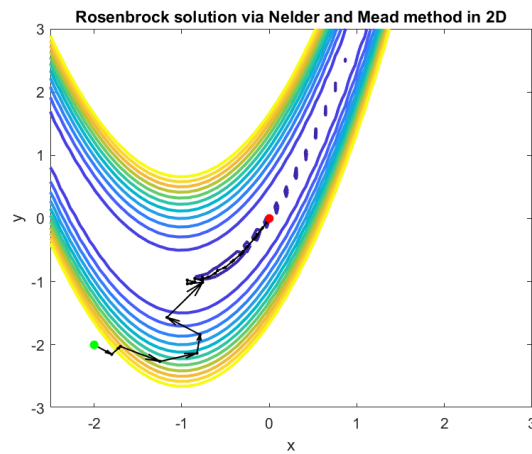
Wymagane wykresy dla punktu $X_0 = (1,0)$:



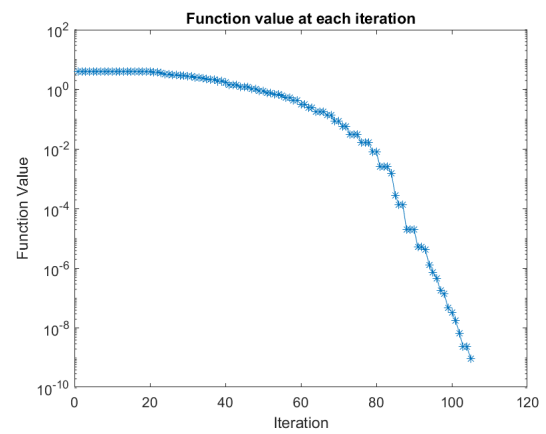
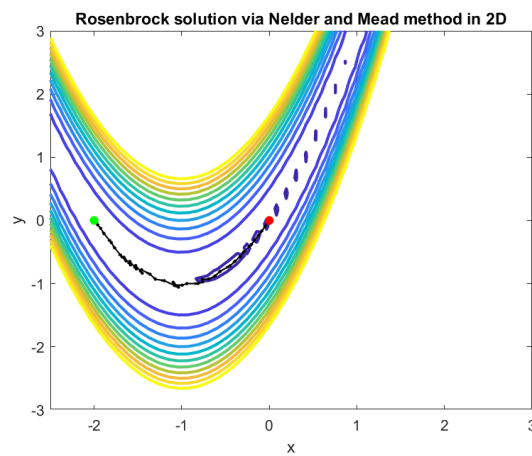
Wymagane wykresy dla punktu $X_0 = (0,-2)$:



Wymagane wykresy dla punktu $X_0 = (-2, -2)$:



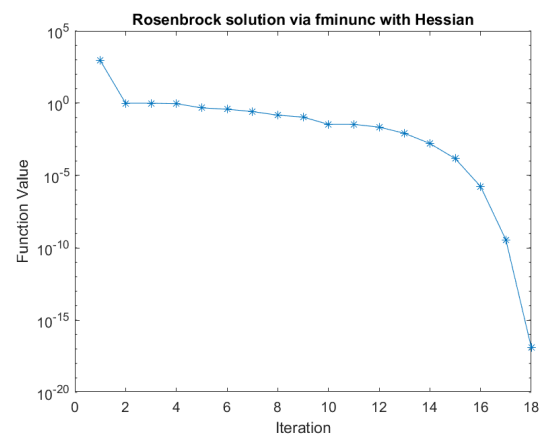
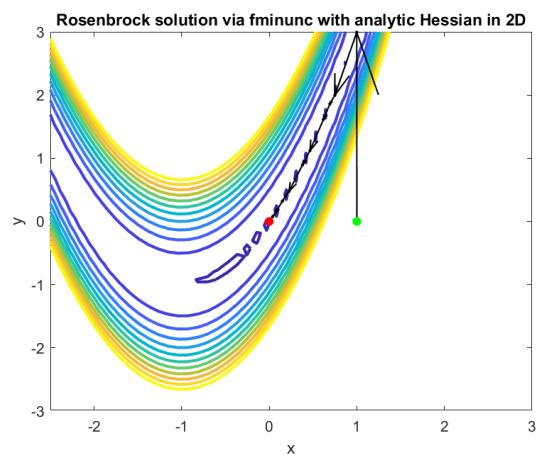
Wymagane wykresy dla punktu $X_0 = (-2, 0)$:



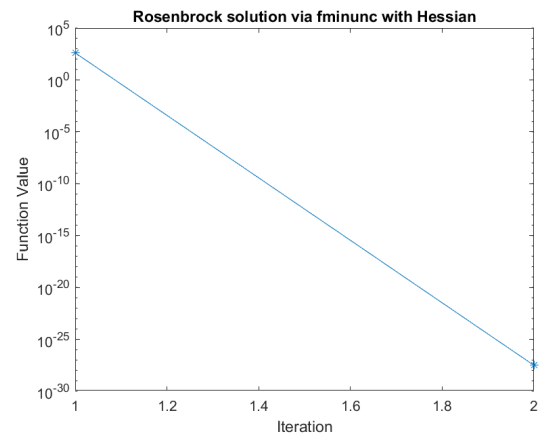
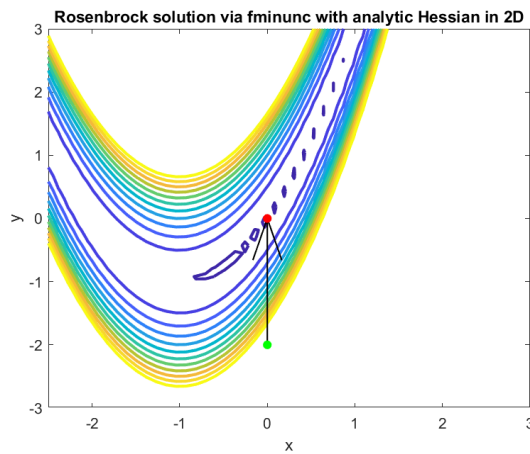
6. Funkcja fminunc - metoda obszaru zaufania - analitycznie obliczona macierz Hessego i gradient.

Punkt Startowy	Punkt końcowy	Liczba iteracji	Dokładność
(1,0)	(3.4141e-09, 6.7014e-09)	17	5.6566e-17
(0,-2)	(-4.4408e-16,-2.6645e-15)	1	7.2969e-30
(-2,-2)	(-1.3774e-09,-2.8063e-09)	25	9.7731e-18
(-2,0)	(-9.3019e-08,-1.9686e-07)	26	4.7408e-14

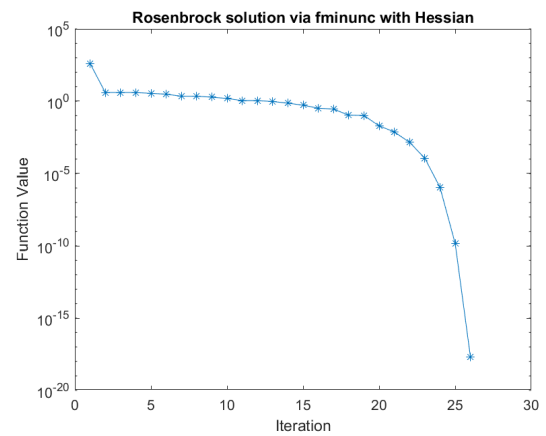
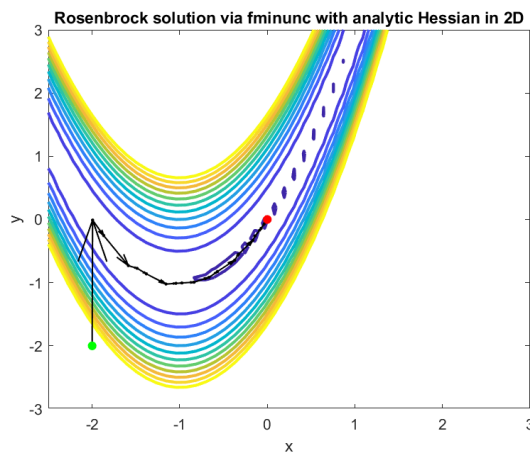
Wymagane wykresy dla punktu $X_0 = (1,0)$:



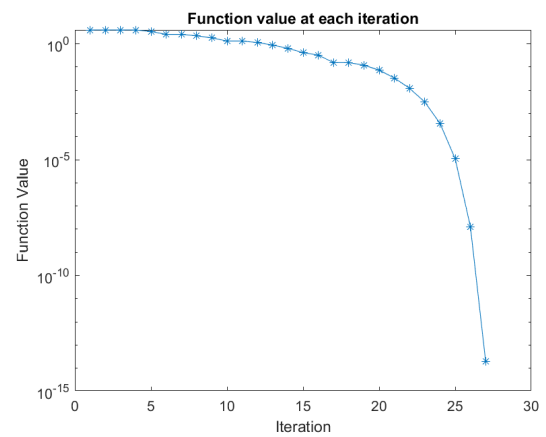
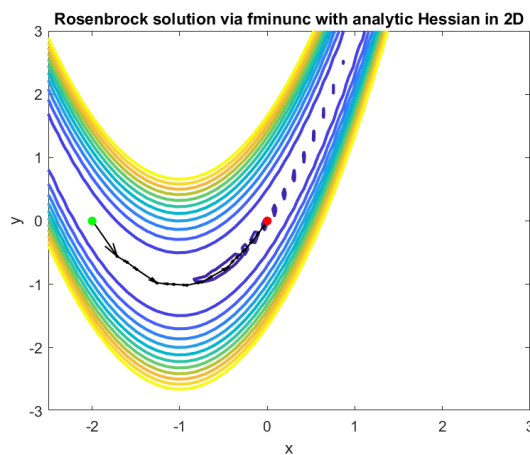
Wymagane wykresy dla punktu $X_0 = (0,-2)$:



Wymagane wykresy dla punktu $X_0 = (-2,-2)$:



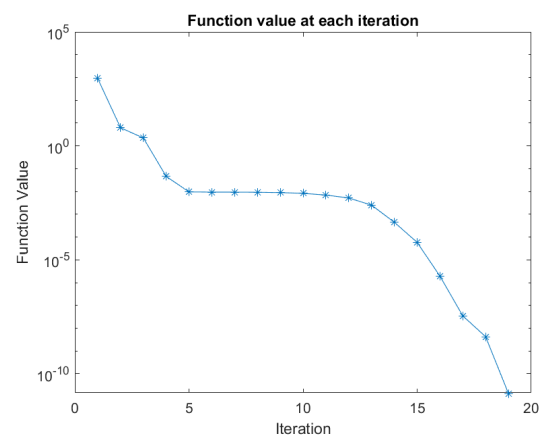
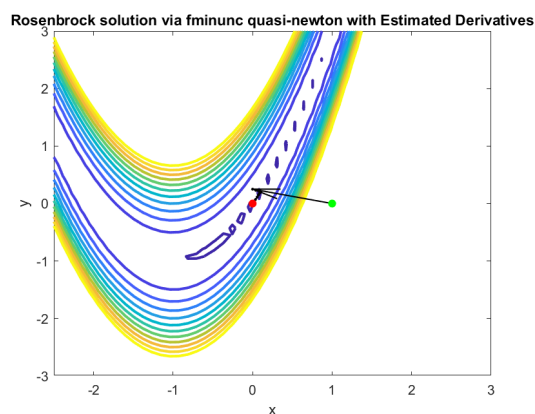
Wymagane wykresy dla punktu $X_0 = (-2,0)$:



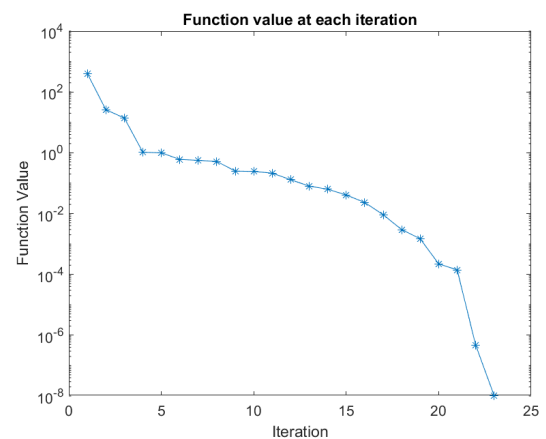
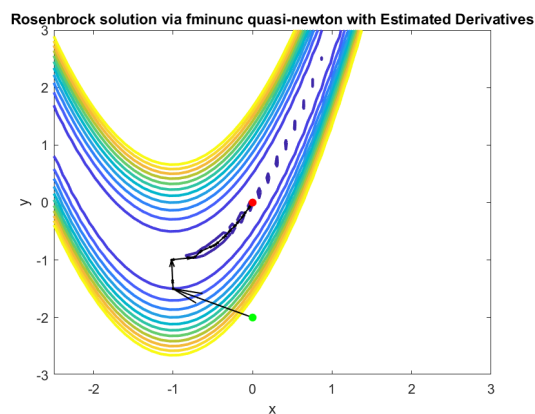
7. Funkcja fminunc - metoda quasi-Newton - estymowany przez algorytm gradient.

Punkt Startowy	Punkt końcowy	Liczba iteracji	Dokładność
(1,0)	(-3.7043e-06,-7.3940e-06)	18	6.8394e-11
(0,-2)	(-10092e-4,-2.0270e-4)	22	5.1271e-08
(-2,-2)	(4.1764e-06,8.1339e-06)	24	8.3603e-11
(-2,0)	(-8.5206e10,-7.9760e-09)	31	6.4343e-17

Wymagane wykresy dla punktu $X_0 = (1,0)$:

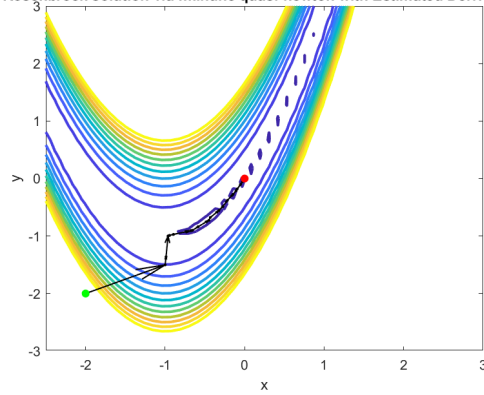


Wymagane wykresy dla punktu $X_0 = (0,-2)$:

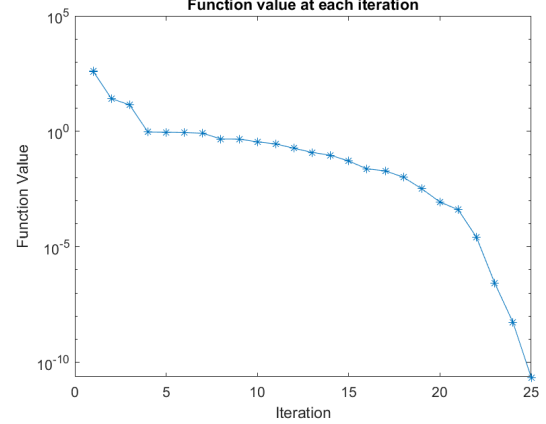


Wymagane wykresy dla punktu $X_0 = (-2, -2)$:

Rosenbrock solution via fminunc quasi-newton with Estimated Derivatives

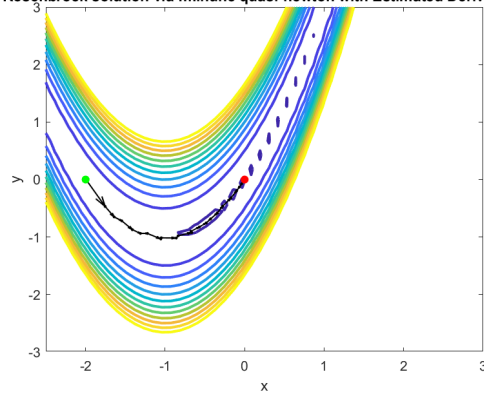


Function value at each iteration

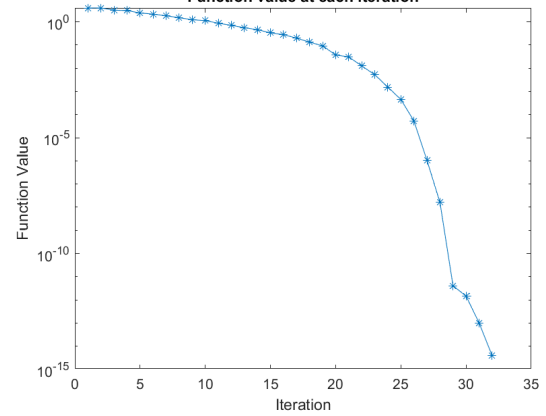


Wymagane wykresy dla punktu $X_0 = (-2, 0)$:

Rosenbrock solution via fminunc quasi-newton with Estimated Derivatives



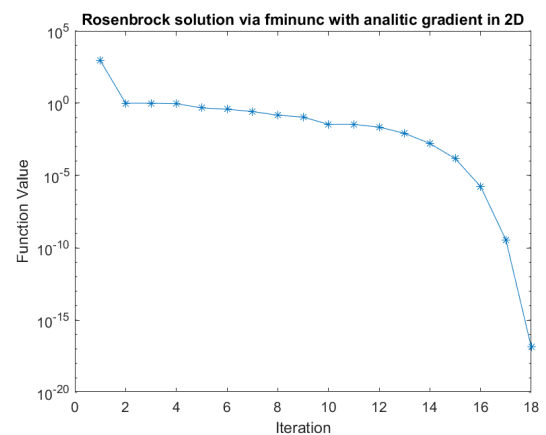
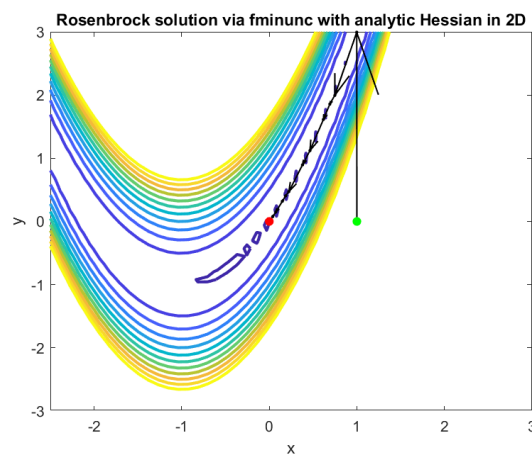
Function value at each iteration



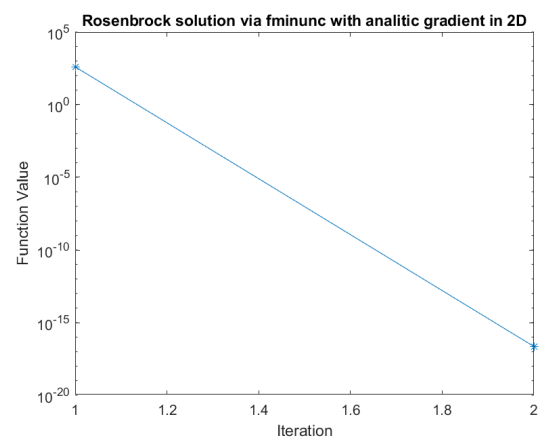
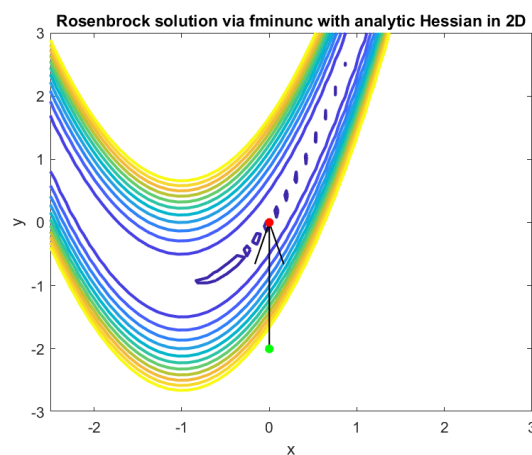
8. Funkcja fminunc - metoda quasi-Newton - estymowany przez algorytm gradient.

Punkt Startowy	Punkt końcowy	Liczba iteracji	Dokładność
(1,0)	(3.4918e-09,6.8564e-09)	17	5.9204e-17
(0,-2)	(4.7564e-09,9.5129e-09)	1	1.1312e-16
(-2,-2)	(-1.4250e-09,-2.9016e-09)	25	1.0450e-17
(-2,0)	(-9.3710e-08, -1.9825e-07)	26	4.8085e-14

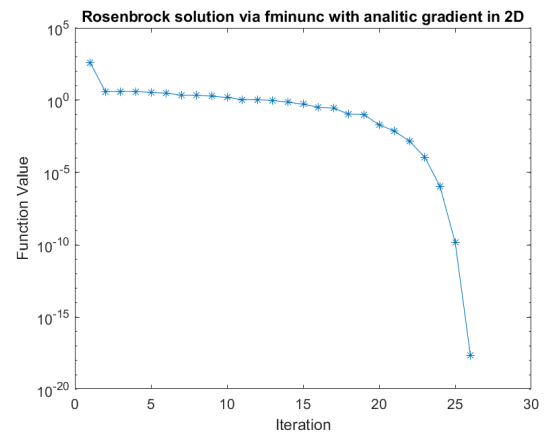
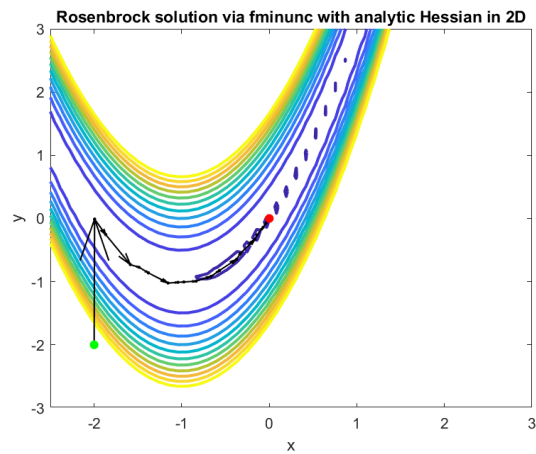
Wymagane wykresy dla punktu $X_0 = (1,0)$:



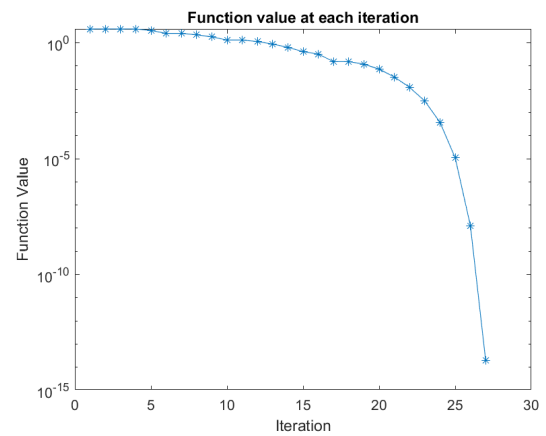
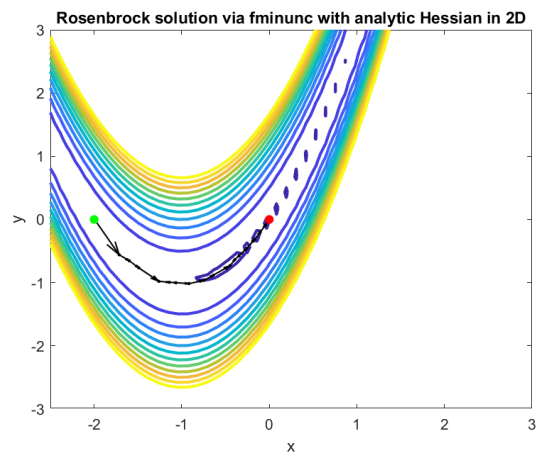
Wymagane wykresy dla punktu $X_0 = (0,-2)$:



Wymagane wykresy dla punktu $X_0 = (-2, -2)$:



Wymagane wykresy dla punktu $X_0 = (-2, 0)$:



9. Ocena działania metod.

Zdecydowanie najlepiej ze wszystkich w tym przykładzie pokazał siebie algorytm *trusted-region*. Wprowadzenie analitycznego wyrażenia zarówno na macierz Hessego, jak i gradientu w porównaniu z metodą wprowadzenia wyrażenia tylko na gradient, nie ma wpływu na szybkość działania optymalizacji (liczba iteracji w każdym z każdego punktu startowego jest taka sama dla obydwu metod), natomiast dokładność się zmienia. Dla metody z wprowadzonym tylko gradientem, dokładność rozwiązania jest mniejsza.

Najgorzej ze wszystkich pokazał siebie algorytm Nelder-Meada. Jest to algorytm bezgradientowy. Jest on najwolniejszy ze wszystkich użytych algorytmów.

Dokładność też okazała się najgorsza.

Algorytm quasi-Newton z estymowanym gradientem i macierzą Hessego pokazał średnie wyniki. Gorsze niż algorytm *trusted-region* i lepsze niż Nelder-Meada

10. References.

MathWorks documentation for Optimization Toolbox(R2022a): Solver - Based Nonlinear Optimization. Retrieved from: <https://www.mathworks.com/help/optim/index.html>

MathWorks documentation for Optimization Toolbox(R2022a): Optimization Toolbox Tutorial. Retrieved from: <https://www.mathworks.com/help/optim/ug/optimization-toolbox-tutorial.html>

MathWorks documentation for Optimization Toolbox(R2022a): Banana Function Minimisation. Retrieved from: <https://www.mathworks.com/help/optim/ug/banana-function-minimization.html>

MathWorks documentation for Graphics 2D and 3D plot(R2022a). Retrieved from: <https://www.mathworks.com/help/matlab/contour-plots-1.html>

Wikipedia."Metoda quasi-Newtona". 18.02.2022 Retrieved from: https://pl.wikipedia.org/wiki/Metoda_quasi-Newtona

Niewiadomska - Szyrkiewicz, E. 2022. Metody optymalizacji nieliniowej [pdf]. https://studia2.elka.pw.edu.pl/file/22L/103B-IBxxx-ISP-AE/priv/ONL_1.pdf