# 1 Opis projektu

Celem projektu jest stworzenie metody simplexu Neldera i Meada . Metoda ta polega na utworzeniu w przestrzeni En+1 *n*-wymiarowego simplexu o *n+1* wierzchołkach tak, aby można było go wpisać w powierzchnię reprezentującą badaną funkcję celu. Jednowymiarowym simplexem jest odcinek o dwóch wierzchołkach, simplexem dwuwymiarowym jest trójkąt i ogólnie simplexem *n*-wymiarowym

o *n+1* wierzchołkach jest zbiór wszystkich punktów określonych przez wektory:



czyli jest to wielościan o *n+1* wierzchołkach rozpiętych na *n+1* wektorach bazowych (Sj). Współrzędne punktów simplexu oznaczono jako xj. Na początku procedury wylicza się współrzędne punktów wierzchołkowych simplexu Pj (dla *j = 1 .. n+1*) przy założeniu pewnej odległości między tymi wierzchołkami (czyli kroku). W następnych iteracjach dokonuje się przekształceń simplexu aż odległość pomiędzy jego wierzchołkami w pobliżu poszukiwanego ekstremum będzie mniejsza od założonej dokładności obliczeń *e*. To właśnie zostało przyjęte jako kryterium zbieżności dla tej metody.

# Algorytm

1. Obliczenie wartości funkcji celu w punktach wierzchołkowych simplexu Fj = f(Pj) dla *j = 1 .. n+1*
2. Wyznaczenie *h* i *L* takich, że f(Ph) = max oraz f(PL) = min spośród zbioru Fj.
3. Obliczenie środka symetrii simplexu P'.
4. Odbicie P\* punktu Ph względem P'.
5. Obliczenie wartości funkcji Fs = f(P') oraz Fo = f(P\*).

Jeśli Fo < min, to:

1. Obliczenie P\*\* (ekspansja) i wartości funkcji Fe = f(P\*\*).
2. Jeśli Fe < max, to podstawiamy Ph = P\*\*, w przeciwnym przypadku podstawiamy Ph = P\*.
3. Powtórzenie procedury od kroku 1), o ile nie jest spełnione kryterium na minimum.

Jeśli Fo > min, to:

1. Jeśli Fo >= f(Pj) dla *j = 1 .. n+1* (z pominięciem *j = h*) i Fo >= max, przejście do następnego kroku. Jeśli zaś Fo < max, to podstawiamy Ph = P\*.
2. Wykonanie kontrakcji P\*\*\* punktu Ph względem P'.
3. Obliczenie Fk = f(P\*\*\*).
4. Jeżeli Fk >= max to wykonujemy redukcję simplexu wg wzoru: Pj = 0,5\*(Pj + PL) dla *j = 1 .. n+1*. Jeżeli natomiast Fk < max, to podstawiamy Ph = P\*\*\* i przechodzimy do kroku 11).
5. Jeśli Fo < f(Pj) dla *j = 1 .. n+1* (z pominięciem *j = h*), podstawiamy Ph = P\*.
6. Powtórzenie procedury od kroku 1), jeśli nie zostało spełnione kryterium na minimum.

Oznaczenia:

x0 - punkt startowy

e - wymagana dokładność obliczeń

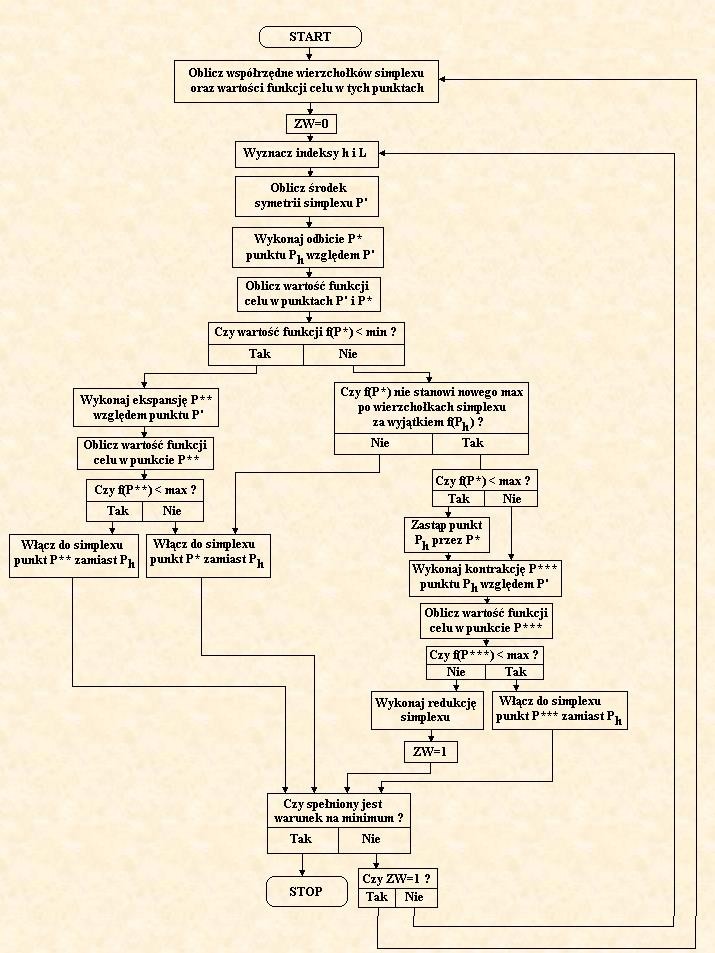
Ph - wybrany punkt wierzchołkowy simplexu spośród *n+1* wierzchołków Pi, w którym wartość badanej funkcji osiąga maksimum.

PL - wybrany punkt wierzchołkowy simplexu spośród *n+1* wierzchołków Pi, w którym wartość badanej funkcji osiąga minimum.

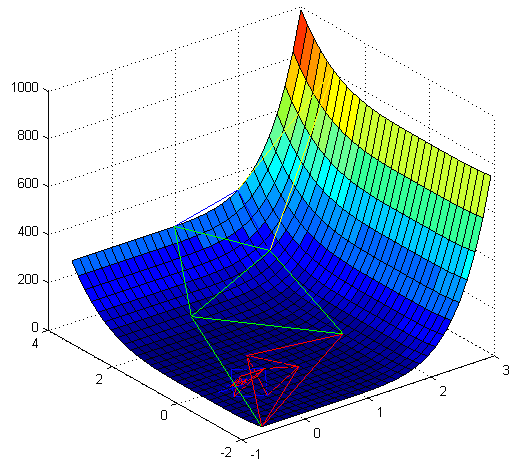
P' - środek symetrii simplexu z wyłączeniem punktu Ph zdefiniowany jako:

P\*- odbicie punktu ' P\*\*-ekspansja punktu P\*\*\*-kontrakcja punktu

# Schemat algorytmu



1. **Przykład**



F(x, y) = x2+y2, punkt początkowy (2, 3)

po 11 iteracjach znaleziono przybliżony pkt. minimalny: (-0,08, -0,07)

legenda: żółty -początkowy symplex, niebieski -odbicie, zielony -ekspansja, czerwony -kontrakcja

1. **Harmonogram**

1)przygotowanie funkcji odbicia, ekspansji i kontrakcji. 2)przygotowanie jednowymiarowego simpleksu

3)przygotowanie dwuwymiarowego simpleksu (połowa projektu)

4)przygotowanie n-wymiarowego simpleksu

5)testowanie funkcji

**6.literatura i inne źródła**

• http://www.scholarpedia.org/article/Nelder-Mead\_algorithm

• https://www.researchgate.net/publication/324084428\_Less\_is\_more\_Simplified\_Nelder-Mead\_method\_for\_large\_unconstrained\_optimization

• https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead\_method

•http://www.kmg.zut.edu.pl/opt/wyklad/bezgrad/simplex.html?fbclid=IwAR0fo4rz1dYzl\_7MzvuruBNySkqW5etm1KVE0XYFHo2eP5ao7hXgkg0POX8