Tema nr. 21 – Algoritmul Muller

Metoda lui Muller este un algoritm de rezolvare a ecuatiilor de forma f(x)=0. Aceasta este bazata pe metoda secantei, care, la fiecare iteratie, construieste o linie printre doua puncte pe graficul functie. Metoda lui Muller, in schimb, foloseste trei puncte aleatoare, printre care traseaza o parabola. Pentru a aproxima radacina, se folosesc intersectiile parabolei cu axa Ox.

Metoda incepe cu 3 valori initiale, la intamplare, x(0), x(-1) si x(-2), cu ajutorul carora se calculeaza prima aproximare, x(1), apoi a doua aproximare, x(2), apoi a treia etc.

Formulele de care ne folosim pentru a ajunge la aproximare sunt:

*q ≡ (xi − xi−*1) / (*xi−*1 *− xi−*2)

*A ≡ qP*(*xi*) *− q*(1 + *q*)*P*(*xi−*1) + *q*2*P*(*xi−*2)

*B ≡* (2*q* + 1)*P*(*xi*) *−* (1 + *q*)2*P*(*xi−*1) + *q*2*P*(*xi−*2)

*C ≡* (1 + *q*)*P*(*xi*)

Iar formula aproximarii este:

*xi*+1 = *xi −* (*xi − xi−*1) \* 2*C / (B ±√(B*2 *−* 4*AC))*

Semnul numitorului este data de valoarea absoluta. A se nota ca trebuie lasata posibilitatea aparitiei unui numitor complex in implementarea acestei probleme. Metoda lui Muller poate fi folosita si pentru aflarea radacinilor functiilor analitice in planul complex.

Solutia gasita implica setarea unei tolerante pana la care se calculeaza aproximarea radacinii, initializarea unui index si a unui bec ajutator.

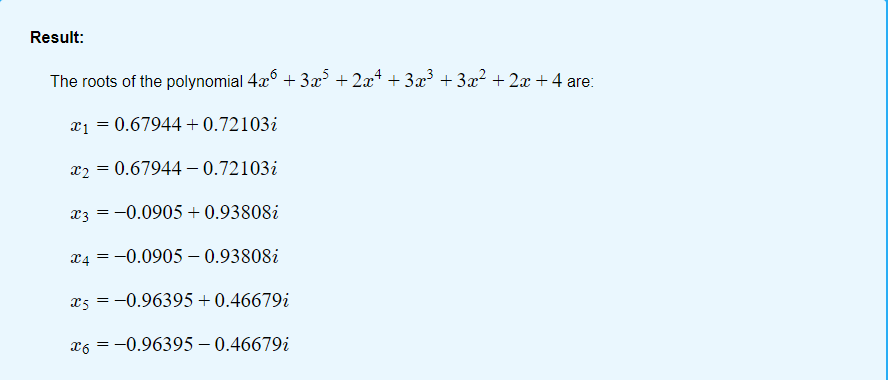
Solutia implica o bucla de tip while pana la aflarea tuturor solutiilor, cand ‘bec’ devine 0 si algoritmul se opreste.

Algoritmul incepe cu verificarea numarului de coloane a vectorului cu coeficienti si rezolvarea cazurilor in care ecuatia are grad cel mult 1. In cazul existentei unei ecuatii de grad mai mare de 2, algoritmul va incepe prin a cere valorile initiale, cu ajutorul carora se traseaza parabola specifica metodei Muller. In cazul neintroducerii a trei valori, se poate apasa direct enter si vor fi folosite trei valori default (a se nota ca valorile alese ar trebui sa fie diferite intre ele). Alegem numarul de iteratii N0 = 100 si incepem calculul aproximarilor radacinilor. Prima data calculam Q, A, B, C folosind o varianta simplificata a formulelor prezentate mai sus, dupa care calculam radicalul formulei pentru aproximare. Verificam valoarea absoluta pentru a determina semnul folosit in cazul numitorului si calculam numitorul, iar mai apoi aproximarea. In cazul in carea valoarea absoluta a acesteia se afla in limitele de toleranta, formam polinomul a carui solutie este chiar aceasta valoare. Afisam valoarea obtinuta, incrementam indexul (care tine evidenta radacinilor obtinute) si introducem in vectorul de radacini valoarea. In cazul in care radacina nu se afla in limitele de toleranta, elementele vectorului cu valori initiale vor fi shiftate la dreapta cu un element, devenind x(-1), x(0) si x(1), unde x(1) este valoarea nou calculata. In cazul in care s-a depasit numarul de iteratii, se afiseaza un mesaj corespunzator si se reia algoritmul cu alte valori initiale introduse.

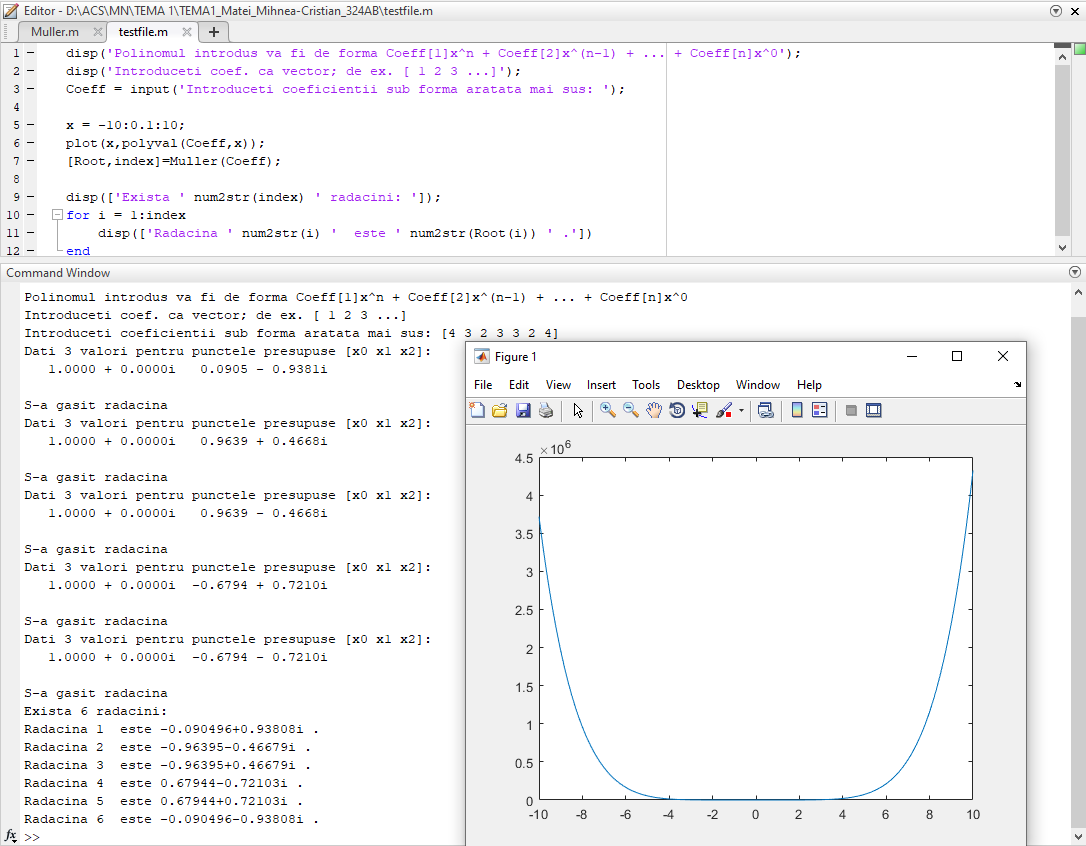
Dupa aflarea unei radacini, cu ajutorul functiei deconv vom imparti polinomul caruia i-am aflat o radacina la polinomul de gradul 1 a carui unica solutie este acea radacina. Restul acestei impartiri va fi introdus in variabila rem.

Rezultate ale algoritmului:

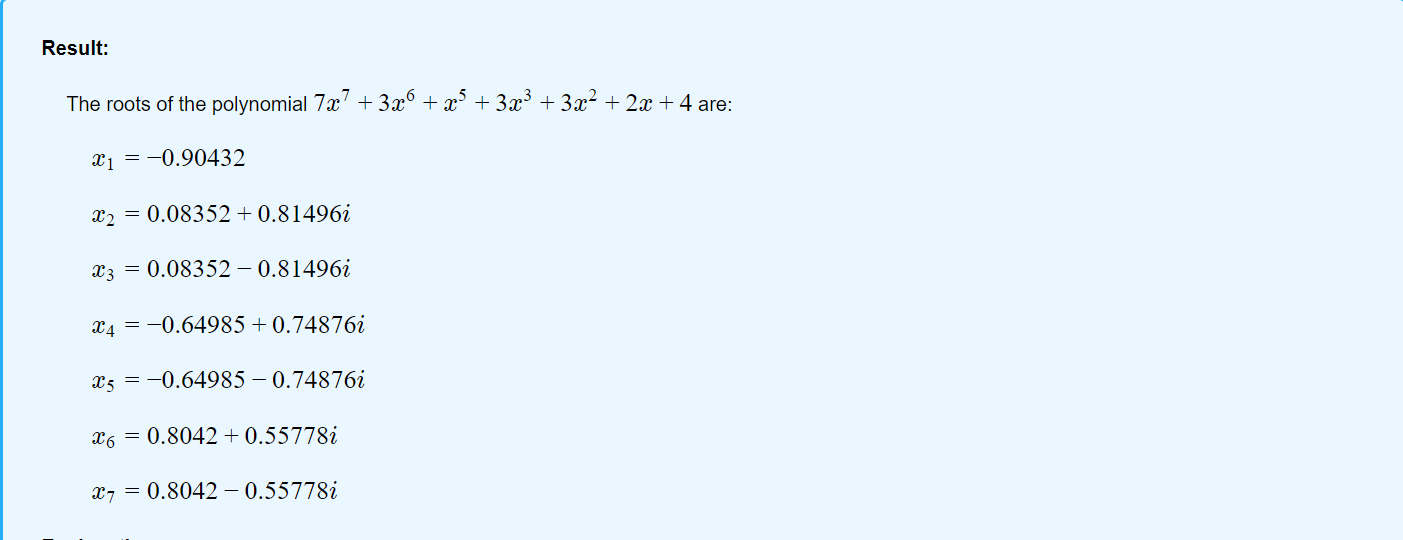
Polinomul 1:



Rezultatul algoritmului:



Polinomul 2:



Rezultatul algoritmului:

