ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

Ακαδ. Έτος: 2018-2019

Τετάρτη, 24/10/2018

Μάθημα: Μέθοδοι Επίλυσης με Η/Υ

Διδάσκοντες: Ν.Δ. Λαγαρός (Αν. Καθηγητής), Αθ. Στάμος (ΕΔΙΠ), Χ. Φραγκουδάκης (ΕΔΙΠ)
Σαββίδης Αμβ. (ΥΔ)

Παραδείγματα για την 3η παράδοση – Μητρώα και συνθήκες στο MATLAB

1. Υπολογισμός βαθμολογίας

Να συνταχθεί συνάρτηση σε Matlab η οποία να δέχεται ως ορίσματα εισόδου το βαθμό γραπτού **bg** και το βαθμό εργασίας **be** και να επιστρέφει τον τελικό βαθμό **bt**, ο οποίος υπολογίζεται ως ο μέσος όρος του βαθμούς της εργασίας και του γραπτού με τις ακόλουθες εξαιρέσεις:

i. Αν ο βαθμός του γραπτού είναι μηδέν τότε ο τελικός βαθμός είναι 0.
ii. Αν ο βαθμός του γραπτού είναι μικρότερος από 3, τότε ο τελικός βαθμός είναι ο μέσος όρος αν ο μέσος όρος είναι μικρότερος από 4, ή διαφορετικά 4.
iii. Αν ο βαθμός της εργασίας είναι μικρότερος από 5, τότε ο τελικός βαθμός είναι ο μέσος όρος αν ο μέσος όρος είναι μικρότερος από 4, ή διαφορετικά 4.

Ο τελικός βαθμός θα στρογγυλεύεται στον πλησιέστερο ακέραιο μέσω της συνάρτησης **round(x)**.

Λύση

Ο κώδικας είναι:

```
function bt=telikos(bg, be)
    if bg == 0
        bt = 0;
    elseif bg < 3
        bt = (bg+be)/2;
        if bt > 4
            bt = 4;
        end
    elseif be < 5
        bt = (bg+be)/2;
        if bt > 4
            bt = 4;
        end
    else
        bt = (bg+be)/2;
    end
bt = round(bt);
end
```

Ελέγχουμε το πρόγραμμα με το script:

```
disp(telikos(0, 9));
      0
disp(telikos(2, 9));
       4
disp(telikos(2, 3));
       3
disp(telikos(9, 2));
```

```
        4
disp(telikos(3, 2));
        3
disp(telikos(8, 9));
        9
```

2. Διερεύνηση δευτεροβάθμιας εξίσωσης

Να υπολογιστούν οι ρίζες της δευτεροβάθμιας εξίσωσης $Ax^2 + Bx + \Gamma = 0$ λαμβάνοντας υπόψη όλες τις δυνατές περιπτώσεις. Οι συντελεστές Α, Β και Γ είναι πραγματικοί αριθμοί.

Λύση

Οι διάφορες περιπτώσεις ανάγονται στο μηδενισμό των διάφορων συντελεστών. Έτσι:

α. Αν Α=Β=0 τότε η εξίσωση δεν έχει νόημα γιατί δεν υπάρχουν άγνωστες ποσότητες.

β. Αν Α=0 και Β≠0 τότε η εξίσωση είναι πρωτοβάθμια και έχει μία λύση: x=-Γ/Β

γ. Αν Α <>0 τότε η εξίσωση είναι δευτεροβάθμια και έχει 1 ή 2 πραγματικές ρίζες, ή 2 συζυγείς μιγαδικές ρίζες. Συγκεκριμένα αν η διακρίνουσα $\Delta = B^2 - 4\,A\,\Gamma$ είναι:

- Δ > 0: Δύο πραγματικές ρίζες: $\dfrac{-B \pm \sqrt{\Delta}}{2A}$
- Δ = 0: Μία πραγματική ρίζα: $\dfrac{-B}{2A}$
- Δ < 0: Δύο συζυγείς μιγαδικές ρίζες: $\dfrac{-B \pm i\sqrt{-\Delta}}{2A}$

Κώδικας

```
function [] = deft(a, b, g)
    if a == 0.0
        if b == 0.0
            disp('ΔΕΝ ΕΙΝΑΙ ΕΞΙΣΩΣΗ');
        else
            disp('ΠΡΩΤΟΒΑΘΜΙΑ ΕΞΙΣΩΣΗ ΡΙΖΑ x=');
            x1 =  -g/b;
            disp(x1);
        end
    else
        d = b^2 - 4.0*a*g;
        if d == 0.0
            disp('ΔΙΠΛΗ ΡΙΖΑ x=');
            x1 = -b/(2.0*a);
            disp(x1);
        elseif d > 0.0
            x1 = (-b + sqrt(d)) / (2.0*a);
            x2 = (-b - sqrt(d)) / (2.0*a);
            disp('ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ x1, x2=');
            disp([x1 x2]);
        else
            x1 = (-b + sqrt(d)) / (2.0*a);
            x2 = (-b - sqrt(d)) / (2.0*a);
            disp('ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ x1, x2=');
            disp([x1 x2]);
        end
    end
end
```

Ελέγχουμε το πρόγραμμα με το script:

```
deft(1, 2, -1);
      ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ x1, x2=
         0.41421  -2.41421deft(1, 2, 1);
deft(1, 2, 1);
      ΔΙΠΛΗ ΡΙΖΑ x=
      -1
deft(1, -2, 3);
      ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ x1, x2=
         1.0000 + 1.4142i   1.0000 - 1.4142i
deft(0, 2, 3);
      ΠΡΩΤΟΒΑΘΜΙΑ ΕΞΙΣΩΣΗ ΡΙΖΑ x=
      -1.5000
deft(0, 0, 3);
      ΔΕΝ ΕΙΝΑΙ ΕΞΙΣΩΣΗ
```

2η λύση

Προηγουμένως η συνάρτηση δεν επιστρέφει τίποτα (κανένα όρισμα εξόδου), διότι όταν A=B=0 δεν υπάρχουν λύσεις για να επιστραφούν. Αν όμως χρειάζονται οι λύσεις στον κώδικα που καλεί τη συνάρτηση, η συνάρτηση πρέπει να επιστρέφει τις λύσεις και με κάποιον τρόπο να ειδοποιεί όταν δεν υπάρχουν. Ένας τρόπος για αυτό είναι, αν δεν υπάρχουν λύσεις, τα ορίσματα εξόδου να πάρουν τιμές που δεν είναι δυνατόν να ισχύουν. Για παράδειγμα μία συνάρτηση που υπολογίζει πλήθος ράβδων οπλισμού, μπορεί να δώσει στο όρισμα εξόδου αρνητική τιμή για να δείξει ότι δεν υπάρχει λύση. Στη δευτεροβάθμια εξίσωση, από μαθηματική άποψη, όλοι οι αριθμοί μπορεί να είναι αποδεκτές λύσεις. Όμως στην επιστήμη του μηχανικού ένα μεγάλος αριθμός όπως 10^{100}, είναι αδύνατον να είναι λύση για οποιοδήποτε πρόβλημα.

Έτσι επαναλαμβάνουμε τον κώδικα με ορίσματα εξόδου, και θεωρούμε επιπλέον ότι οι μιγαδικές λύσεις δεν είναι αποδεκτές λύσεις.

Κώδικας

```
function [x1, x2] = deft(a, b, g)
    if a == 0.0
        if b == 0.0
            x1 = 1e100;
            x2 = 1e100;
        else
            x1 =  -g/b;
            x2 = 1e100;
        end
    else
        d = b^2 - 4.0*a*g;
        if d == 0.0
            x1 = -b/(2.0*a);
            x1 = x2;
        elseif d > 0.0
            x1 = (-b + sqrt(d)) / (2.0*a);
            x2 = (-b - sqrt(d)) / (2.0*a);
        else
            x1 = 1e100;
            x2 = 1e100;
        end
    end
end
```

3. Πίνακας στροφών

Να υπολογιστούν και να τυπωθούν τα στοιχεία του πίνακα στροφών με δεδομένα τις γωνίες ω, φ, κ σε μοίρες.

$$R=\begin{bmatrix} \cos\varphi\cos\kappa & \cos\omega\sin\kappa+\sin\omega\sin\varphi\cos\kappa & \sin\omega\sin\kappa-\cos\omega\sin\varphi\cos\kappa \\ \cos\varphi\sin\kappa & \cos\omega\cos\kappa-\sin\omega\sin\varphi\sin\kappa & \sin\omega\cos\kappa+\cos\omega\sin\varphi\sin\kappa \\ sin\varphi & -sin\omega\, cos\varphi & cos\omega\, cos\varphi \end{bmatrix}$$

Λύση

Ο υπολογισμός των τριγωνομετρικών συναρτήσεων γίνεται σε rad. Ο υπολογισμός κάθε τριγωνομετρικής συνάρτησης γίνεται μία φορά για λόγους ταχύτητας και αναγνωσιμότητας. Οι ονομασίες μεταβλητών επιλέγονται έτσι ώστε η ονομασίες να αντικατοπτρίζουν τις τιμές που αποθηκεύουν οι μεταβλητές.

Κώδικας

```
function R=strof(w, f, ak)
    w = w * pi / 180.0;
    f = f * pi / 180.0;
    ak = ak * pi / 180.0;
    cf = cos(f);
    sf = sin(f);
    cw = cos(w);
    sw = sin(w);
    ck = cos(ak);
    sk = sin(ak);
    R = [cf * ck    cw * sk + sw * sf * ck    sw * sk - cw*sf*ck;
         sf * sk    cw * ck - sw * sf * sk    sw * ck + cw * sf * sk;
         sf        -sw * cf                   cw * cf                 ];
    disp(R);
end
```

4. Υπολογισμός τροχιάς τοπογραφικού δορυφόρου

Η σύνταξη τοπογραφικού από δορυφορικές μετρικές εικόνες απαιτεί τη γνώση της θέσης του δορυφόρου X(t), Y(t), Z(t) σε κάθε χρονική στιγμή t. Χρησιμοποιώντας συνδυασμό GPS και αδρανειακού συστήματος πλοήγησης, υπολογίζονται οι συντεταγμένες X_P, Y_P, Z_P και X_Q, Y_Q, Z_Q και οι ταχύτητες U_P, V_P, W_P και U_Q, V_Q, W_Q σε δύο χρονικές στιγμές t_P και t_Q αντίστοιχα. Για τυχαία χρονική στιγμή t μεταξύ t_P και t_Q η θέση του δορυφόρου X(t), Y(t), Z(t) μπορεί να υπολογιστεί:

- Με γραμμική παρεμβολή: $X(t) = A_1+B_1t$, $Y(t) = A_2+B_2t$, $Z(t) = A_3+B_3t$
- Με κυβική παρεμβολή: $X(t) = A_1+B_1t+C_1t^2+D_1t^3$, $Y(t) = A_2+B_2t+C_2t^2+D_2t^3$, $Z(t) = A_3+B_3t+C_3t^2+D_3t^3$

α) Να συνταχθεί συνάρτηση σε Matlab που με δεδομένα τις δύο χρονικές στιγμές t_P και t_Q και τις αντίστοιχες συντεταγμένες να υπολογίζει τους συντελεστές της γραμμικής παρεμβολής.

β) Να συνταχθεί συνάρτηση σε Matlab που με δεδομένα τις δύο χρονικές στιγμές t_P και t_Q και τις αντίστοιχες συντεταγμένες και ταχύτητες να υπολογίζει τους συντελεστές της κυβικής παρεμβολής.

γ) Να συνταχθεί script που να υπολογίζει και να τυπώνει τη θέση του δορυφόρου ανά 0.5 sec μεταξύ t_P και t_Q με γραμμική και κυβική παρεμβολή. Από την εφημερίδα του δορυφόρου δίνονται:


```
<stateVec maneuver="NO" num="2" qualInd="1">
  <timeUTC>2009-02-02T04:19:25.000000</timeUTC>
  <timeGPS>917583580</timeGPS>
  <timeGPSFraction>0.0000000000000000E+00</timeGPSFraction>
  <posX>4.51806707262837235E+06</posX>
  <posY>2.75808651607648935E+06</posY>
  <posZ>4.39935303662791569E+06</posZ>
  <velX>4.96953799999999956E+03</velX>
  <velY>1.07864899999999989E+03</velY>
  <velZ>-5.76280299999999988E+03</velZ>
</stateVec>
<stateVec maneuver="NO" num="3" qualInd="1">
  <timeUTC>2009-02-02T04:19:35.000000</timeUTC>
  <timeGPS>917583590</timeGPS>
  <timeGPSFraction>0.0000000000000000E+00</timeGPSFraction>
  <posX>4.56749465524839889E+06</posX>
  <posY>2.76866907010982186E+06</posY>
  <posZ>4.34145693145494256E+06</posZ>
  <velX>4.91586899999999969E+03</velX>
  <velY>1.03785400000000004E+03</velY>
  <velZ>-5.81630100000000039E+03</velZ>
```


Λύση

Για τη γραμμική παρεμβολή και την τετμημένη X ισχύει:

$X_P = A_1+B_1t_P \Rightarrow X_P = 1 \cdot A_1 + t_P \cdot B_1$

$X_Q = A_1+B_1t_Q \Rightarrow X_Q = 1 \cdot A_1 + B_1 \cdot t_Q$

Οι 2 σχέσεις αποτελούν ένα σύστημα 2 εξισώσεων με 2 αγνώστους τα A και B, που μπορεί να επιλυθεί με το Matlab. Ομοίως για την τεταγμένη Y και την κατηγμένη Z.

Για τη κυβική παρεμβολή και την τετμημένη X ισχύει:

$X_P = A_1+B_1t_P+C_1t_P^2+D_1t_P^3 \Rightarrow X_P = 1 \cdot A_1 + t_P \cdot B_1 + t_P^2 \cdot C_1 + t_P^3 \cdot D_1$

$X_Q = A_1+B_1t_Q+C_1t_Q^2+D_1t_Q^3 \Rightarrow X_Q = 1 \cdot A_1 + t_Q \cdot B_1 + t_Q^2 \cdot C_1 + t_Q^3 \cdot D_1$

Παίρνοντας την πρώτη παράγωγο της τετμημένης ως προς το χρόνο υπολογίζεται η ταχύτητα, για την οποία ισχύει:

$U_P = B_1+2C_1t_P+3D_1t_P^2 \Rightarrow X_P = 0 \cdot A_1 + 1 \cdot B_1 + 2t_P \cdot C_1 + 3t_P^2 \cdot D_1$

$U_Q = B_1+2C_1t_Q+3D_1t_Q^2 \Rightarrow X_Q = 0 \cdot A_1 + 1 \cdot B_1 + 2t_Q \cdot C_1 + 3t_Q^2 \cdot D_1$

Οι 2 σχέσεις της τετμημένης και οι 2 σχέσεις της ταχύτητας U αποτελούν ένα σύστημα 4 εξισώσεων με 4 αγνώστους τα A, B, C, D που μπορεί να επιλυθεί με το Matlab. Ομοίως για τη διεύθυνση Y και τη διεύθυνση Z.

<u>Υπολογισμός συντελεστών γραμμικής παρεμβολής και συντεταγμένων</u>

Θέτουμε C_p=[X_P, Y_P, Z_P], C_Q=[X_Q, Y_Q, Z_Q] και:

$$E_i = \begin{bmatrix} A_i \\ B_i \end{bmatrix} \quad , \quad F = \begin{bmatrix} 1 & t_P \\ 1 & t_Q \end{bmatrix} \quad , \quad G = \begin{bmatrix} C_P(i) \\ C_Q(i) \end{bmatrix}$$ όπου i μπορεί να είναι 1 ή 2 ή 3,

Κώδικας:

```
function [AA, BB] = gram1(tp, cp, tq, cq, i)
    F = [1 tp; 1 tq];
    G = [cp(i) cq(i)]';
    E = F\G;
    AA = E(1);
    BB = E(2);
end

function [A, B]=gram(tp, cp, tq, cq)
    [A(1), B(1)] = gram1(tp, cp, tq, cq, 1);
    [A(2), B(2)] = gram1(tp, cp, tq, cq, 2);
    [A(3), B(3)] = gram1(tp, cp, tq, cq, 3);
end

function [X, Y, Z] = gramxyz(A, B, t)
    X = A(1)+B(1)*t;
    Y = A(2)+B(2)*t;
    Z = A(3)+B(3)*t;
end
```

<u>Υπολογισμός συντελεστών γραμμικής παρεμβολής και συνεταγμένων</u>

Θέτουμε C_p=[X_P, Y_P, Z_P], C_Q=[X_Q, Y_Q, Z_Q], T_p=[U_P, V_P, W_P], S_Q=[U_Q, V_Q, W_Q], και:

$$E_i = \begin{bmatrix} A_i \\ B_i \\ C_i \\ D_i \end{bmatrix} \quad , \quad F = \begin{bmatrix} 1 & t_P & t_P^2 & t_P^3 \\ 1 & t_Q & t_Q^2 & t_Q^3 \\ 0 & 1 & 2t_P & 3t_P^2 \\ 0 & 1 & 2t_Q & 3t_Q^2 \end{bmatrix} \quad , \quad G = \begin{bmatrix} C_P(i) \\ C_Q(i) \\ S_P(i) \\ S_Q(i) \end{bmatrix}$$ όπου i μπορεί να είναι 1 ή 2 ή 3,

Κώδικας:

```
function [AA, BB, CC, DD] = kyb1(tp, cp, sp, tq, cq, sq, i)
    F = [1 tp tp^2 tp^3; 1 tq tq^2 tq^3; 0 1 2*tp 3*tp^2; 0 1 2*tq 3*tq^2];
    G = [cp(i) cq(i) sp(i) sq(i)]';
    E = F\G;
    AA = E(1);
    BB = E(2);
    CC = E(3);
    DD = E(4);
end

function [A, B, C, D]=kyb(tp, cp, sp, tq, cq, sq)
```

```
    [A(1), B(1), C(1), D(1)] = kyb1(tp, cp, sp, tq, cq, sq, 1);
    [A(2), B(2), C(2), D(2)] = kyb1(tp, cp, sp, tq, cq, sq, 2);
    [A(3), B(3), C(3), D(3)] = kyb1(tp, cp, sp, tq, cq, sq, 3);
end

function [X, Y, Z] = kybxyz(A, B, C, D, t)
    X = A(1)+t.*(B(1)+ t.*(C(1) + t.*D(1)));
    Y = A(2)+t.*(B(2)+ t.*(C(2) + t.*D(2)));
    Z = A(3)+t.*(B(3)+ t.*(C(3) + t.*D(3)));
end
```

<u>Κύριο πρόγραμμα</u>

Κώδικας:

```
clear; clc; close all;
tp = 25.000000;
cp = [4.51806707262837235E+06
      2.75808651607648935E+06
      4.39935303662791569E+06];
sp = [4.96953799999999956E+03
      1.07864899999999989E+03
     -5.76280299999999988E+03];
tq = 35.000000;
cq = [4.56749465524839889E+06
      2.76866907010982186E+06
      4.34145693145494256E+06];
sq = [4.91586899999999969E+03
      1.03785400000000004E+03
     -5.81630100000000039E+03];

[A, B]=gram(tp, cp, tq, cq);
t = tp:0.5:tq;
[X, Y, Z] = gramxyz(A, B, t);
temp = [X' Y' Z'];
disp('Linear:');
disp(temp);

[A, B, C, D]=kyb(tp, cp, sp, tq, cq, sq);
[X, Y, Z] = kybxyz(A, B, C, D, t);
temp = [X' Y' Z'];
disp('Cubic:');
disp(temp);
```

<u>Έλεγχος προγράμματος</u>

Τρέξιμο του προγράμματος δίνει:

```
Linear:
   4.5181e+06   2.7581e+06   4.3994e+06
   4.5205e+06   2.7586e+06   4.3965e+06
   4.5230e+06   2.7591e+06   4.3936e+06
   4.5255e+06   2.7597e+06   4.3907e+06
   4.5280e+06   2.7602e+06   4.3878e+06
   4.5304e+06   2.7607e+06   4.3849e+06
   4.5329e+06   2.7613e+06   4.3820e+06
   4.5354e+06   2.7618e+06   4.3791e+06
```

4.5378e+06	2.7623e+06	4.3762e+06
4.5403e+06	2.7628e+06	4.3733e+06
4.5428e+06	2.7634e+06	4.3704e+06
4.5453e+06	2.7639e+06	4.3675e+06
4.5477e+06	2.7644e+06	4.3646e+06
4.5502e+06	2.7650e+06	4.3617e+06
4.5527e+06	2.7655e+06	4.3588e+06
4.5551e+06	2.7660e+06	4.3559e+06
4.5576e+06	2.7666e+06	4.3530e+06
4.5601e+06	2.7671e+06	4.3501e+06
4.5626e+06	2.7676e+06	4.3472e+06
4.5650e+06	2.7681e+06	4.3444e+06
4.5675e+06	2.7687e+06	4.3415e+06

Cubic:

4.5181e+06	2.7581e+06	4.3994e+06
4.5206e+06	2.7586e+06	4.3965e+06
4.5230e+06	2.7592e+06	4.3936e+06
4.5255e+06	2.7597e+06	4.3907e+06
4.5280e+06	2.7602e+06	4.3878e+06
4.5305e+06	2.7608e+06	4.3849e+06
4.5330e+06	2.7613e+06	4.3820e+06
4.5354e+06	2.7618e+06	4.3792e+06
4.5379e+06	2.7624e+06	4.3763e+06
4.5404e+06	2.7629e+06	4.3734e+06
4.5428e+06	2.7634e+06	4.3705e+06
4.5453e+06	2.7640e+06	4.3676e+06
4.5478e+06	2.7645e+06	4.3647e+06
4.5503e+06	2.7650e+06	4.3618e+06
4.5527e+06	2.7655e+06	4.3589e+06
4.5552e+06	2.7661e+06	4.3560e+06
4.5577e+06	2.7666e+06	4.3531e+06
4.5601e+06	2.7671e+06	4.3502e+06
4.5626e+06	2.7676e+06	4.3473e+06
4.5650e+06	2.7681e+06	4.3444e+06
4.5675e+06	2.7687e+06	4.3415e+06