ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
Τμήμα Επιστήμης Υπολογιστών

ΗΥ-215: Εφαρμοσμένα Μαθηματικά για Μηχανικούς
Εαρινό Εξάμηνο 2021-22
Διδάσκοντες: Γ. Στυλιανού, Γ. Καφεντζής

Δεύτερη Σειρά Ασκήσεων

Ημερομηνία Ανάθεσης: 1/3/2022 Ημερομηνία Παράδοσης: 15/3/2022, 15:45

Οι ασκήσεις με [$\star$] είναι **bonus**, $+10$ μονάδες η καθεμία στο βαθμό αυτής της σειράς ασκήσεων (δηλ. μπορείτε να πάρετε μέχρι $100/80$ σε αυτή τη σειρά.)

Άσκηση 1 - Σήματα

(α΄) Ένα σήμα λέγεται *άρτιο* αν ισχύει ότι

$$x(t) = x(-t) \tag{1}$$

ενώ λέγεται *περιττό* αν

$$x(t) = -x(-t) \tag{2}$$

Ελέγξτε αν τα παρακάτω σήματα είναι άρτια, περιττά, ή τίποτε από τα δυο.

i. $x(t) = t^3$
ii. $x(t) = t^3|t|$
iii. $x(t) = |t^3|$
iv. $x(t) = \frac{1}{2}(e^{jt} + e^{-jt})$
v. $x(t) = 1 + \sin(2\pi t)$

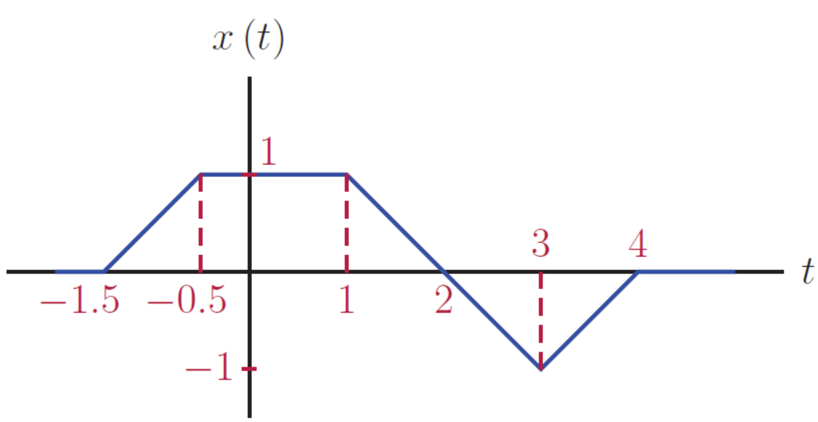
(β΄) Επαληθεύστε ή διαψεύστε τις παρακάτω προτάσεις, δικαιολογώντας επαρκώς ή βρίσκοντας αντιπαραδείγματα:

i. Το άθροισμα δυο σημάτων ισχύος είναι πάντα σήμα ισχύος.
ii. Το γινόμενο δυο σημάτων ισχύος μπορεί να είναι ένα σήμα ενέργειας.

Απ.: (α) i. περιττό, ii. περιττό, iii. άρτιο, iv. άρτιο, v. τίποτα, (β) i. Λάθος, ii. Σωστό

Άσκηση 2 - Μετασχηματισμοί Σημάτων

Έστω το σήμα του σχήματος 1. Σχεδιάστε τα παρακάτω σήματα:



Σχήμα 1: *Σχήμα Άσκησης 2.*

(α΄) $x(-t+3)$

(β΄) $x(-t)$

(γ΄) $x\left(\frac{t-1}{3}\right)$

(δ΄) $x(4t-3)$

Άσκηση 3 - Συναρτήσεις Δέλτα

Απλοποιήστε τις παρακάτω εκφράσεις

(α΄) $\dfrac{(t^2+1)\delta(t+1)}{t^2+9}$

(β΄) $\dfrac{\sin(at)\delta(t)}{t}$

(γ΄) $\displaystyle\int_{-\infty}^{+\infty} e^{t-1}\cos(\pi(t-5)/2)\delta(t-3)dt$

(δ΄) $\displaystyle\int_{-\infty}^{+\infty} \delta(2t-3)\sin(\pi t)dt$

(ε΄) $\displaystyle\int_{t}^{\infty} (\tau^2+1)\delta(\tau+2)d\tau$

(ϛ΄) $\displaystyle\int_{0}^{2} e^{j2t}\delta(t-4)dt$

Άσκηση 4 - Συναρτήσεις Δέλτα και Βηματικές

Υπολογίστε και σχεδιάστε τη γενικευμένη παράγωγο του σήματος

$$g(t) = 3\cos(2\pi t)\mathrm{rect}\left(t-\frac{1}{2}\right) \tag{3}$$

Αξιοποιήστε τη γραφή του τετραγωνικού παλμού με χρήση δυο βηματικών συναρτήσεων.

Απ.: $g'(t) = -6\pi\sin(2\pi t)\mathrm{rect}\left(t-\frac{1}{2}\right) + 3\delta(t) - 3\delta(t-1)$

Άσκηση 5 - Συστήματα και Ιδιότητες

Ελέγξτε αν τα παρακάτω συστήματα είναι γραμμικά, χρονικά αμετάβλητα, ευσταθή, αιτιατά, και δυναμικά.

(α΄) $y(t) = |x(t)| + x(t+1)$

(β΄) $y(t) = tx(t)$

Απ:

	Γρ.	Χ.Α.	Ευστ.	Αιτ.	Δυν.
(α)	✗	✓	✓	✗	✓
(β)	✓	✗	✗	✓	✗

Άσκηση 6 - Διαφορικές Εξισώσεις ως Συστήματα

Υπολογίστε τη συνολική έξοδο του συστήματος που περιγράφεται από τη διαφορική εξίσωση

$$\frac{d^2}{dt^2}y(t) + \frac{d}{dt}y(t) - 6y(t) = x(t) + 2\frac{d}{dt}x(t) \tag{4}$$

με αρχικές συνθήκες $y(0^-) = 0$, $y'(0^-) = -1$ και είσοδο $x(t) = e^{-3t}u(t)$. Δηλαδή

(α΄) βρείτε την απόκριση μηδενικής εισόδου, $y_{zi}(t)$

(β΄) βρείτε την κρουστική απόκριση, $h(t)$

(γ΄) βρείτε την απόκριση μηδενικής κατάστασης, $y_{zs}(t)$

(δ΄) τέλος, βρείτε τη συνολική έξοδο $y_{tot}(t) = y_{zi}(t) + y_{zs}(t)$

Απ.: (α) $y_{zi}(t) = \left(\frac{1}{5}e^{-3t} - \frac{1}{5}e^{2t}\right)u(t)$, (β) $h(t) = (e^{-3t} + e^{2t})u(t)$, (γ) $y_{zs}(t) = [te^{-3t} + \frac{1}{5}e^{2t} - \frac{1}{5}e^{-3t}]u(t)$

Άσκηση 7 - Ολοκλήρωση στο MATLAB/Octave

Στην προσπάθειά σας να προσομοιώσετε (και φυσικά να κατανοήσετε) τα σήματα και τις έννοιες που θα δείτε στο μάθημα αυτό, θα χρειαστείτε να υπολογίζετε ολοκληρώματα στο MATLAB/Octave. Ευτυχώς η διαδικασία είναι πολύ απλή μέσω του ολοκληρώματος Riemann που (ελπίζουμε να ☺) γνωρίζετε από τον Απειροστικό Λογισμό. Ας δούμε πως μπορούμε να υπολογίσουμε ένα ολοκλήρωμα στο MATLAB/Octave.

Ολοκλήρωμα Riemann

Ο B. Riemann πρότεινε την προσέγγιση ενός ολοκληρώματος από τμηματικά σταθερές συναρτήσεις, των οποίων το αθροιστικό εμβαδό δίνει μια τιμή για το ολοκλήρωμα, δηλ.

$$\int_a^b x(t)dt = \lim_{\Delta t \to 0} \sum_{i=1}^{N} x(\Delta t_i)\Delta t \tag{5}$$

Τι μας λέει η παραπάνω σχέση; Μας λέει ότι η τιμή ενός ορισμένου ολοκληρώματος μπορεί να βρεθεί αν κάνουμε τα εξής:

(α΄) Χωρίζουμε το διάστημα $[a, b]$ σε N ίσα μέρη μήκους Δt το καθένα. Αυτή η διαμέριση του διαστήματος λέγεται *ομοιόμορφη*.

(β΄) Παίρνουμε ένα σημείο Δt_i σε κάθε ένα από αυτά τα διαστήματα. Για παράδειγμα, το μέσο του διαστήματος Δt ή όποιο άλλο θέλουμε.

(γ΄) Υπολογίζουμε την τιμή της συνάρτησης $x(t)$ στα σημεία Δt_i.

(δ΄) Έχουμε τώρα N τιμές της συνάρτησης $x(t)$ και N τμήματα μήκους Δt.

(ε΄) Πολλαπλασιάζουμε τις N τιμές της συνάρτησης $x(t)$, δηλ. τις τιμές $x(\Delta t_i)$, με τα N το πλήθος Δt, και τις προσθέτουμε όλες μαζί, για να πάρουμε το αποτέλεσμα.

Για να είναι ακριβές το αποτέλεσμα, πρέπει το Δt να είναι όσο μικρότερο γίνεται, δηλ. ιδανικά να ισχύει $\Delta t \to 0$, ώστε να υπάρχει ισότητα μεταξύ ολοκληρώματος και αθροίσματος.

Θυμηθείτε από την προηγούμενη σειρά ασκήσεων ότι ο συνεχής χρόνος δεν μπορεί να αναπαρασταθεί ακριβώς στο MATLAB/Octave, οπότε χρειάζεται να πάρουμε *δείγματα* αυτού και να κάνουμε τις πράξεις μας. Όσο πιο πολλά τα δείγματα, τόσο πιο κοντά θα είναι η προσέγγισή μας στην πραγματική τιμή του ολοκληρώματος.

Ας υπολογίσουμε τώρα το (διάσημο) ολοκλήρωμα

$$\int_0^{\pi/2} \left(\frac{t}{\sin(t)}\right)^2 dt = \pi \ln(2) \tag{6}$$

μέσω της προσέγγισης του Riemann.

Αρχικά πρέπει να προσέξουμε ότι η συνάρτηση μέσα στο ολοκλήρωμα δεν ορίζεται για $t = 0$. Οπότε:

```
% 8a orisoume ton a3ona apo th xronikh stigmh Dt ws to pi/2, oxi apo to t=0 giati
% ekei o paronomasths den orizetai
% O a3onas tmhmatopoieitai ana Dt arketa mikro alla ths epiloghs mas
```

```
Dt = 0.001;
t = Dt:Dt:pi/2;

% Orizoume th synarthsh mas
x = (t./sin(t)).^2;

% Ypologizoume to a8roisma Riemann
Result = Dt * sum(x)
```

και το MATLAB/Octave αποκρίνεται ως

```
Result =

    2.1764
```

Ας το επιβεβαιώσουμε:

```
% Epibebaiwsh
pi*log(2)

ans =

    2.1776
```

Η διαφορά οφείλεται στη δειγματοληψία του άξονα t. Αν μειώσουμε το βήμα μας $Dt = 0.001$ σε (π.χ.) $Dt = 0.00001$, θα πλησιάσουμε ακόμα περισσότερο στο ακριβές αποτέλεσμα.

Μια πολύ σπουδαία ιδιότητα που έχει το MATLAB/Octave είναι η ικανότητά του να εκτελεί (μερικώς) *συμβολικούς* υπολογισμούς, δηλ. υπολογισμούς χωρίς αριθμητικές τιμές! Ας δούμε πως θα υλοποιούσαμε το ολοκλήρωμα

$$\int_0^{2\pi} \frac{dt}{\frac{5}{4} - \cos(t)} = \frac{8\pi}{3} \tag{7}$$

όχι αριθμητικά αυτή τη φορά, αλλά συμβολικά[1].

```
% Orizoume mia symbolikh metablhth t
syms t;

% Orizoume th synarthsh mas
x = 1./(5/4 - cos(t));

% Zhtame apo th symbolikh synarthsh "int" na ypologisei gia mas
% to oloklhrwma!!
Result = int(x, t, 0, 2*pi)
```

και το MATLAB/Octave μας δίνει

```
Result =

(8*pi)/3
```

Βλέπετε ότι όχι μόνο μας υπολογίζει το σωστό αποτέλεσμα αλλά μας το δίνει και σε κλειστή μορφή!!

Με βάση τα παραπάνω, επιβεβαιώστε στο MATLAB/Octave τα ολοκληρώματα **τόσο αριθμητικά όσο και συμβολικά**:

[1] Οι χρήστες Octave θα χρειαστούν το πακέτο symbolic, όπως στην 1η Σειρά Ασκήσεων.

I. $\displaystyle\int_0^{\pi} \sin(4\theta)\cos(5\theta)d\theta = -\frac{8}{9}$

II. $\displaystyle\int_0^{2\pi} \frac{dt}{\frac{5}{4} - \cos(t)} = \frac{8\pi}{3}$

III. $\displaystyle\int_{-\infty}^{+\infty} e^{-t^2} dt = \sqrt{\pi}$

IV. $\displaystyle\int_1^{2} \frac{(t+2)(t-2)}{t^2} dt = -1$

Όπου εμπλέκεται το άπειρο ως άκρο ολοκληρώματος, μπορείτε να χρησιμοποιήσετε ένα "μεγάλο" άξονα t για τον αριθμητικό υπολογισμο και τη συμβολική μεταβλητή inf για το συμβολικό υπολογισμό.

Παραδώστε μόνο τον κώδικα που εκτελεί τους υπολογισμούς.

Άσκηση 8 - Ανίχνευση Δραστηριότητας Ομιλίας στο πρότυπο GSM - MATLAB/Octave

Μόνο και μόνο με τις έννοιες της *ενέργειας* και των απλών πρότυπων σημάτων, όπως ο *τετραγωνικός παλμός*, μπορείτε να δείτε μερικές εφαρμογές όπως η *ανίχνευση δραστηριότητας ομιλίας - voice activity detection*. Αλγόριθμοι που υλοποιούν τέτοιες εφαρμογές υπάρχουν σε κάθε κινητό τηλέφωνο. Μια ιδιαίτερα απλοϊκή υλοποίηση ενός τέτοιου είναι η εξής:

- Διαβάζετε ένα αρχείο ομιλίας στο MATLAB/Octave που σας δίνεται.
- Παραθυροποιείτε με χρήση τετραγωνικού παλμού το σήμα σας, "κόβοντας " έτσι ένα τμήμα του.
- Για κάθε παραθυροποιημένο σήμα, υπολογίζετε την ενέργεια του από τη γνωστή σχέση.
- Ανάλογα με την τιμή της ενέργειας, αποφασίζετε αν το παραθυροποιημένο σήμα περιέχει ομιλία ή όχι.
- Προχωράτε στο επόμενο παραθυροποιημένο σήμα και επαναλαμβάνετε, ως το τέλος του σήματος.

Ερωτήματα που προκύπτουν από την ανάγνωση του παραπάνω αλγορίθμου:

(α΄) **Πόση είναι η διάρκεια του τετραγωνικού παλμού;** Μια τιμή κοντά στα 30 ms είναι καλή. Δεν υπάρχει βέλτιστη διάρκεια, υπάρχουν trade-offs είτε έχει μεγάλη είτε μικρή διάρκεια.

(β΄) **Πόση απόσταση θα έχουν τα διαδοχικά παραθυροποιημένα σήματα μεταξύ τους;** Στην πράξη θα πρέπει όχι απλά να είναι διαδοχικά, αλλά να έχουν και κάποιο ποσοστό *επικάλυψης*. Για τους δικούς μας σκοπούς, θα χρησιμοποιήσουμε 0% επικάλυψη, δηλ. τα παραθυροποιημένα σήματα θα είναι ακριβώς διαδοχικά, χωρίς "κενά" μεταξύ τους (αφού κάτι τέτοιο θα σήμαινε απώλεια εκτίμησης δραστηριότητας ομιλίας).

(γ΄) **Πως αποφασίζω με βάση την ενέργεια αν το εκάστοτε παραθυροποιημένο σήμα είναι τμήμα ομιλίας;** Αυτή είναι μια ερώτηση που δέχεται αρκετές απαντήσεις - προσεγγίσεις. Ξανά μια απλή προσέγγιση είναι ότι στην αρχή του σήματος, υπάρχουν μερικά δευτερόλεπτα σιωπής - μπορείτε να εκτιμήσετε την ενέργεια σε αυτά τα παραθυροποιημένα σήματα και να αποφασίζετε βάσει αυτής (με κάποιο στατιστικό) για τα επόμενα τμήματα ομιλίας.

Θα χρησιμοποιήσουμε το ολοκλήρωμα Riemann για να υπολογίσουμε την ενέργεια - δείτε ξανά την Άσκηση 7 σε αυτή τη σειρά. Το σήμα ομιλίας που έχουμε ηχογραφήσει και σας δίνουμε είναι *ψηφιακό* και όταν το διαβάζετε στο MATLAB/Octave, το μετατρέπετε σε *διακριτού χρόνου*. Έστω ότι το σήμα ομιλίας είναι το σήμα

```
x = [1, 0.5, 1.2, -1.7, 2.4, -0.3, -1.1, 0.6, -0.9];
```

Οι τιμές που βλέπετε αποτελούν *δείγματα* του σήματος ομιλίας συνεχούς χρόνου που καταγράφηκε. Τα δείγματα αυτά λήφθηκαν ανά $1/F_s$ δευτερόλεπτα πραγματικού χρόνου, με F_s να είναι η *συχνότητα δειγματοληψίας*, που σας δίνεται έτοιμη κάθε φορά που ανοίγετε ένα αρχείο ήχου στο MATLAB/Octave. Αν, για παράδειγμα, η συχνότητα δειγματοληψίας είναι 16000 Hz, τότε το παραπάνω σήμα 9 δειγμάτων προέρχεται από ένα σήμα συνεχούς χρόνου διάρκειας $9/16000 = 562.5$ μ$s = 0.5625$ ms $= 0.0005625$ s - πολύ μικρό. Για να φτάσουμε στα 30 ms που αναφέρει παραπάνω η άσκηση, χρειαζόμαστε 480 δείματα - αφού $480/16000 = 0.03$ s $= 30$ ms. Ας συνεχίσουμε όμως σε αυτό το toy example κι ας υποθέσουμε όμως ότι θέλουμε να υπολογίσουμε την ενέργεια ανά 187.5 μs, δηλ. ανά μόλις 3 δείγματα. Τότε θα χωρίσουμε το σήμα σε τριάδες, που ουσιαστικά αυτό αντιστοιχεί σε διαδοχικούς πολλαπλασιασμούς με μη επικαλυπτόμενους τετραγωνικούς παλμούς μοναδιαίου πλάτους και διάρκειας 187.5 μs: προκύπτουν τρεις τριάδες, $(1, 0.5, 1.2), (-1.7, 2.4, -0.3), (-1.1, 0.6, -0.9)$, και θα υπολογίσουμε την ενέργειά τους όπως στην Άσκηση 7, δηλ.

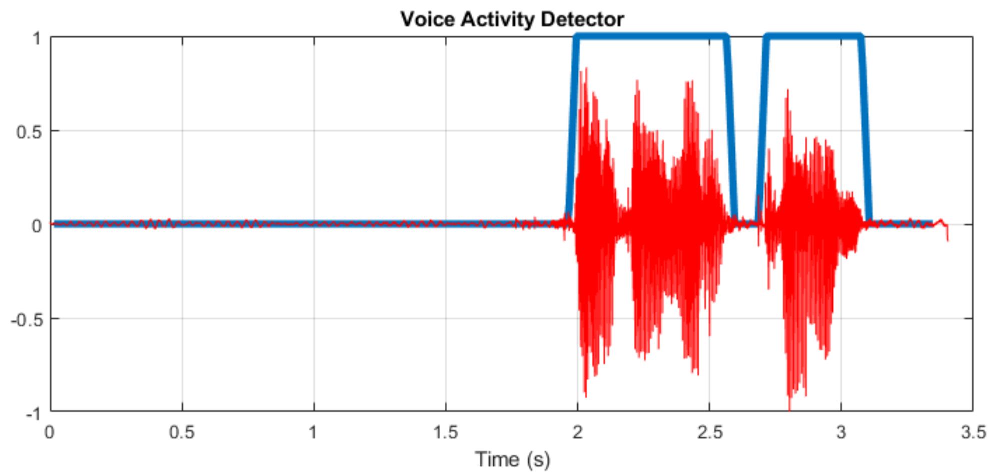
```
Dt = 1/Fs;
E1 = Dt*sum([1, 0.5, 1.2].^2);
E2 = Dt*sum([-1.7, 2.4, -0.3].^2);
E3 = Dt*sum([-1.1, 0.6, -0.9].^2);
```

Σας γνωστοποιούμε ότι τα πρώτα 2 δευτερόλεπτα του σήματος περιέχουν σιωπή. Ακολουθώντας τον παρακάτω σκελετό, υλοποιήστε έναν απλό ενεργειακό ανιχνευτή δραστηριότητας ομιλίας.

```
% Fortwsh arxeiou sto MATLAB
[s, Fs] = audioread('speech.wav');

% 8eloume 30 ms tmhmata, se posa deigmata antistoixoun?
T = % INSERT CODE;

% Synolikh diarkeia shmatos se deigmata
L = length(s);

% Posoi palmoi xwrane sth diarkeia L tou shmatos?
N = % INSERT CODE;

% Diatre3e to shma
for i = 1:N-1
    % Kopse to katallhlo tmhma (kane xrhsh tou i kai tou T)
    windowed_speech = % INSERT CODE;

    % Ypologise thn energeia tou tmhmatos
    Energy(i) = % INSERT CODE;

    % Kentro tou para8yropoihmenou shmatos
    Time(i) = round(% INSERT CODE);
end

% Katwfli apofashs (p.x.)
Number_of_Silence_Frames = round(2*Fs/T);
Threshold = mean(Energy(1:Number_of_Silence_Frames));

% Apofash
for i = 1:N-1
    if Energy(i) > Threshold
        D(i) = 1; % Voice detected
    else
        D(i) = 0; % Voice not detected
    end
end
```

```
% Grafhma
t = 0:1/Fs:(L-1)/Fs;
n = 0:L-1;
Din = interp1(Time, D, n, "linear");
plot(t, Din, "LineWidth", 4); hold on; plot(t, s/max(abs(s)), 'r');
hold off; grid;
title('Voice Activity Detector'); xlabel('Time (s)');
```

Για ευκολία, ο παραπάνω κώδικας σας δίνεται και στο αρχείο `vad.m`, ενώ το αρχείο `speech.wav` θα το βρείτε επίσης στο site του μαθήματος. Αν τα κάνετε όλα σωστά, ο κώδικας θα σας επιστρέψει ένα σχήμα πολύ κοντά στο Σχήμα 2.



Σχήμα 2: *Αποτέλεσμα Ανιχνευτή Δραστηριότητας Ομιλίας.*

Παραδώστε μόνο τον συμπληρωμένο κώδικα - προαιρετικά μπορείτε να παραδώσετε τυπωμένο και το plot που σας επιστρέφει.

[⋆] **Άσκηση 9 - Δώστε μας τη φωνή σας ☺**

Ηχογραφήστε τον εαυτό σας να εκφωνεί μια μικρή πρόταση της επιλογής σας, διάρκειας 2-3 δευτερολέπτων. Κρατήστε τουλάχιστον δυο δευτερόλεπτα σιωπής πριν μιλήσετε. Χρησιμοποιήστε οποιοδήποτε software της επιλογής σας για την ηχογράφηση (προτάσεις: Wavesurfer, Audacity). Προσπαθήστε η ηχογράφηση να είναι όσο γίνεται πιο καθαρή (αποφύγετε το μικρόφωνο του laptop σας, χρησιμοποιήστε εξωτερικό μικρόφωνο ή μικρόφωνο από headset). Καλό είναι οι προδιαγραφές της ηχογράφησης - τις οποίες θα ορίσετε από το πρόγραμμα ηχρογράφησης - να είναι οι ακόλουθες:

- Συχνότητα δειγματοληψίας: 16000 Hz
- Ακρίβεια: 16-bit

Αποθηκεύστε την ηχογράφησή σας σε μορφή .WAV. Χρησιμοποιήστε τον ανιχνευτή δραστηριότητας ομιλίας από την Άσκηση 8 πάνω στη δική σας ομιλία![2]

Παραδώστε μόνο το plot που σας επιστρέφει.

[2]ΜΗ μοιραστείτε μεταξύ σας κάποιο κοινό αρχείο ηχογράφησης: εύκολα μπορούμε να το ανιχνεύσουμε από το plot και θα θεωρηθεί αντιγραφή, με όλες τις συμφωνημένες συνέπειες.

[$\star$] **Άσκηση 10 - Διαφορικές εξισώσεις στο MATLAB/Octave**

Οι διαφορικές εξισώσεις περιγράφουν πραγματικά πολλά φυσικά φαινόμενα ή μηχανικές κατασκευές στον πραγματικό κόσμο. Όμως για να προσομοιώσουμε διαφορικές εξισώσεις στον υπολογιστή μας (και να προβλέψουμε πράγματα πριν τα υλοποιήσουμε στην πραγματικότητα) πρέπει να τις "διακριτοποιήσουμε": δεν μπορούμε να αποθηκεύσουμε στον υπολογιστή μας τη συνεχή μεταβλητή t και τις τιμές των συναρτήσεων που εμπλέκονται στη διαφορική εξίσωση. Πρέπει λοιπόν να επιλέξουμε κάποιες τιμές για τη μεταβλητή t, όπως κάναμε στις γραφικές παραστάσεις της προηγούμενης σειράς ασκήσεων. Φυσικά το MATLAB/Ocave έχει έτοιμες συναρτήσεις που λύνουν πολύπλοκες διαφορικές εξισώσεις αλλά δε θα μας απασχολήσουν σε αυτήν την άσκηση. Θα ακολουθήσουμε μια πιο απλή κι ελέγξιμη προσέγγιση.

Έστω η διαφορική εξίσωση

$$\frac{d}{dt}y(t) + 4y(t) = 4x(t) \tag{8}$$

με $y(0^-) = 0$, που περιγράφει (π.χ.) ένα κύκλωμα RC. Μια απλή μέθοδος διακριτοποίησης της διαφορικής εξίσωσης είναι η μέθοδος του Euler[3]. Σκεφτείτε ότι η παράγωγος μιας συνάρτησης $x(t)$ δεν είναι κάτι περισσότερο από το λόγο $\Delta x/\Delta t$ με το $\Delta t \to 0$. Μπορούμε λοιπόν να γράψουμε ότι

$$\frac{d}{dt}y(t) + 4y(t) = 4x(t) \tag{9}$$

$$\lim_{\Delta t\to 0}\frac{\Delta y}{\Delta t} + 4y(t) = 4x(t) \tag{10}$$

$$\lim_{h\to 0}\frac{y(t+h)-y(t)}{h} + 4y(t) = 4x(t) \tag{11}$$

Παρατηρήστε ότι το h είναι η χρονική διαφορά μεταξύ δυο στιγμών που λαμβάνουμε τιμές για τη συνάρτηση $y(t)$. Μπορούμε να θέσουμε λοιπόν ως $h = T_s$ (και $t = nT_s$, $n \in \mathbb{N}$), με T_s να είναι η *περίοδος δειγματοληψίας* στην προσπάθειά μας να διακριτοποιήσουμε τη διαφορική εξίσωση, δηλ. μια τιμή που καθορίζει κάθε πότε παίρνουμε τιμές από τις συναρτήσεις μας. Θέλουμε φυσικά αυτό το T_s να είναι αρκετά μικρό - όσο μικρό επιθυμούμε. Έτσι γράφουμε

$$\frac{d}{dt}y(t) + 4y(t) = 4x(t) \tag{12}$$

$$\lim_{h\to 0}\frac{y(t+h)-y(t)}{h} + 4y(t) = 4x(t) \tag{13}$$

$$\lim_{T_s\to 0}\frac{y((n+1)T_s)-y(nT_s)}{T_s} + 4y(nT_s) = 4x(nT_s) \tag{14}$$

και αν θέλουμε να εκφράσουμε την εξίσωση κατ΄ ευθείαν στο διακριτό χρόνο, τότε

$$y((n+1)T_s) = y(nT_s) - 4T_s y(nT_s) + 4T_s x(nT_s) \tag{15}$$

Βρήκαμε λοιπόν μια σχέση που διακριτοποιεί τη διαφορική εξίσωση. Μπορούμε να γράψουμε κώδικα που για διάφορες τιμές του T_s μας δίνει μια προσέγγιση της λύσης της διαφορικής εξίσωσης. Σας δίνεται ότι η λύση της διαφορικής εξίσωσης είναι

$$y(t) = (1 - e^{-4t})u(t) \tag{16}$$

για είσοδο $x(t) = u(t)$, και ο παρακάτω κώδικας που πρέπει να συμπληρώσετε. Ο παρακάτω κώδικας εξετάζει πως συμπεριφέρεται η προσέγγισή μας σε σχέση με την τιμή της μεταβλητής T_s, ενώ υπολογίζει και το επί τοις εκατό σφάλμα μεταξύ της πραγματικής λύσης και της προσέγγισής μας, σε κάθε χρονική στιγμή nT_s:

$$Error(nT_s) = \frac{y_{true}(nT_s) - y_{approx}(nT_s)}{y_{true}(nT_s)} \times 100 \tag{17}$$

Ακολουθεί ο κώδικας.

[3] Πάλι αυτός!

```
Ts = % INSERT CODE HERE   % Periodos deigmatolhpsias
t = 0:Ts:2;               % Dianysma xronou t [0,2]
x = ones(size(t));        % Eisodos x(t) = u(t)

% Theorhtikh lysh diaforikhs (t einai 8etiko e3' orismou)
y_true = % INSERT CODE HERE

% Proseggistikh lysh
y = zeros(size(t));       % Desmeysh mnhmhs

y(1) = 0;                 % Arxikh timh y(0) (to MATLAB/Octave ari8mei apo to 1)
for n = 1:length(y)-1     % For loop gia ylopoihsh diaforikhs
    y(n+1) = % INSERT CODE HERE
end

% Grafhmata
clf; subplot(211);
plot(t, y_true ,'-', t, y, 'r.-'); grid;
title('Exact and approximate solutions for RC circuit');
xlabel('Time (s)');
ylabel('Amplitude');
legend('Exact solution', 'Approximate solution', 'Location', 'SouthEast');

% Ypologismos sfalmatos se pososto epi tois ekato
err_pct = (y - y_true)./(y_true)*100;
subplot(212);
plot(t(2:length(t)), err_pct(2:length(t)), 'r.-'); grid
title('Percent approximation error');
xlabel('Time (sec)');
ylabel('Error (%)');
```

Για ευκολία, ο παραπάνω κώδικας σας δίνεται και στο αρχείο `Ex10code.m`. Συμπληρώστε τις γραμμές που λείπουν κι επιλέξτε δυο τιμές για τη μεταβλητή `Ts`: 0.1 και 0.01. Τι παρατηρείτε στα γραφήματα που προκύπτουν σε κάθε περίπτωση;

Παραδώστε τον συμπληρωμένο κώδικα MATLAB/Octave - μόνο για την τιμή $T_s = 0.1$. Καταγράψτε όμως τις παρατηρήσεις/απαντήσεις σας στο παραπάνω ερώτημα σε σχόλιο στον κώδικα.