

Αλγόριθμοι σχεδιασμού βασικών σχημάτων με το Matlab

Στο σημερινό εργαστήριο θα ασχοληθούμε με την υλοποίηση με την βοήθεια το Matlab των αλγορίθμων που διδάχθηκαν στη θεωρία για τον σχεδιασμό ευθύγραμμου τμήματος, κύκλου και έλλειψης.

Πρέπει να σημειώσουμε από τώρα ότι το Matlab δεν μας δίνει την δυνατότητα να φωτίσουμε συγκεκριμένα pixel στην οθόνη μας αφού το σχεδιαστικό της υπόβαθρο δεν έχει δημιουργηθεί για αυτό τον σκοπό. Οπότε δεν πρέπει να μπερδεύουμε την εντολή **plot** που βλέπουμε στην θεωρία με την αντίστοιχη εντολή σχεδιασμού στο Matlab.

Εμείς για να πετύχουμε την ψευδαίσθηση ότι δουλεύουμε με pixel θα κάνουμε χρήση της εντολής **image** που είδαμε την χρήση της στο προηγούμενο εργαστήριο.

Επίσης όταν θα γίνεται αναφορά στις function :

bresline , **circle** και **ellipse** θα αναφερόμαστε στα αντίστοιχα m-files που μπορείτε να τα βρείτε στην eclass στα έγγραφα στον φάκελο με όνομα **m-files**.

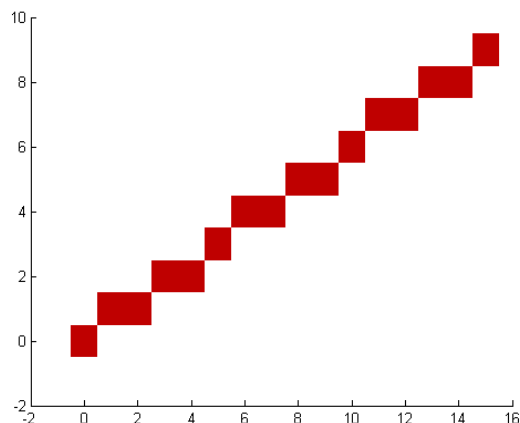
Σχεδιασμός Ευθύγραμμου Τμήματος

Στην θεωρία είδαμε τον αλγόριθμο για τον σχεδιασμό ευθύγραμμου τμήματος που ανήκει στο 1ο οκταμώριο του επιπέδου. Η υλοποίηση του σε Matlab είναι η συνάρτηση **bresline**.

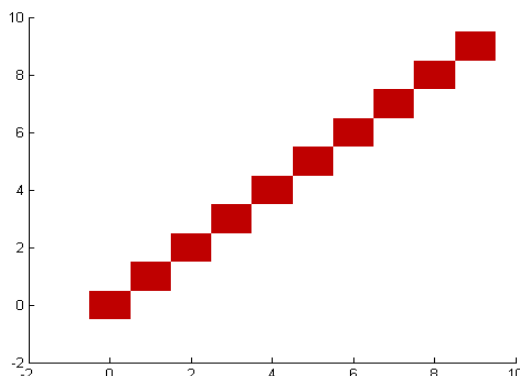
Την οποία μπορούμε να την καλέσουμε για παράδειγμα

```
>> bresline(0,0,15,9,60)
```

Όπου (0,0) είναι το σημείο P₁ και (15,9) το σημείο P₂ (60 είναι μια τιμή για τον χρωματισμό). Το αποτέλεσμα θα είναι ο σχεδιασμός της οθόνης μας του τμήματος P₁P₂



Αν όμως πούμε να σχεδιαστεί το >> **bresline(0,0,9,15,60)** το αποτέλεσμα που θα πάρουμε θα είναι η ακόλουθη εικόνα



Που προφανώς είναι λάθος αφού μπορεί να ξεκινάει από το (0,0) αλλά καταλήγει στο (9,9). Αυτό συμβαίνει γιατί το ευθύγραμμο τμήμα (0,0) – (9,15) ανήκει στο δεύτερο οκταμόριο. Κάτι που ο αλγόριθμος μας δεν μπορεί να το αντιμετωπίσει.

Για αυτό πρέπει και εμείς να μετατρέψουμε ανάλογα τον αλγόριθμο για να καλύπτει όλες τις δυνατές περιπτώσεις.

Στην θεωρία είδαμε κάποιους τρόπους για να μπορέσουμε να καλύψουμε τις περιπτώσεις ευθυγράμμων τμημάτων σε όλο το επίπεδο ανεξαρτήτου οκταμορίου.

Είναι σημαντικό να σημειώσουμε ότι οι περιπτώσεις των τμημάτων που είναι είτε κάθετα είτε οριζόντια στους άξονες (δηλαδή αυτά που τα P_1, P_2 έχουν είτε ίδιο x είτε ίδιο y πρέπει να τα δούμε χωριστά)

Μια καλή ιδέα είναι να παρατηρήσουμε ότι τα οκταμόρια έχουν κάποια σχέση μεταξύ τους.

Για παράδειγμα ότι αν έχω ένα τμήμα P_1P_2 που ανήκει στο 5ο οκταμόριο αν το δω σαν τμήμα P_2P_1 αυτό ανήκει στο 1ο οκταμόριο. Επομένως μπορώ κάθε φορά να επιλέγω τα P_1, P_2 με την σειρά που εγώ θέλω έτσι ώστε πάντα να βρίσκομαι στα 4 πρώτα οκταμόρια. Μετά αρκεί να δω κάθε φορά ποιά είναι η κατεύθυνση σχεδιασμού και να προχωρήσω αντίστοιχα.

Μια άλλη λύση είναι να προχωρήσω ακόμα περισσότερο και να καταλήξω στο τέλος να κάνω τον σχεδιασμό με βάση μόνο τα 2 πρώτα οκταμόρια του σχήματος. Ένας τέτοιος τρόπος περιγράφηκε και στην θεωρία και βασίζεται σε επιπλέον συμμετρίες που μπορούμε να διακρίνουμε και να τις εκμεταλλευτούμε ανάλογα.

Καμιά φορά μπορεί να ζητηθεί αντί να σχεδιάσουμε το ευθύγραμμο τμήμα απλά να υπολογίσουμε έναν πίνακα με τα σημεία που πρέπει αν “φωτιστούν”. Αυτό γίνεται πολύ εύκολα με μια τροποποίηση του αρχικού function όπως μπορούμε να δούμε παρακάτω.

```
function A = bresline2(x1,y1,x2,y2)
A= [];
dx = x2-x1;
dy = y2-y1;
x = x1;
y = y1;
c1 = 2*dy;
error = 2*dy-dx ;
c2 = error-dx;
while x <= x2
    A= [A;[x y]];
    x = x+1;
    if error < 0
        error = error + c1;
    else
        y = y+1;
        error = error + c2;
    end
end
end
```

Το function αυτό μας επιστρέφει τον πίνακα A που έχει τόσες γραμμές όσα τα σημεία του τμήματος και στην πρώτη στήλη είναι οι συντεταγμένες x και στην δεύτερη οι αντίστοιχες y των σημείων.

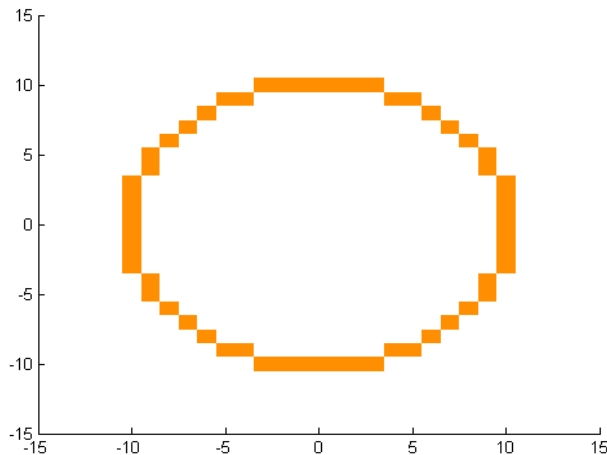
Αν θέλω τώρα να σχεδιάσω τα σημεία του A αρκεί να γράψω

```
hold on
for i=1:size(A,1)
image(A(i,1),A(i,2),60)
end
```

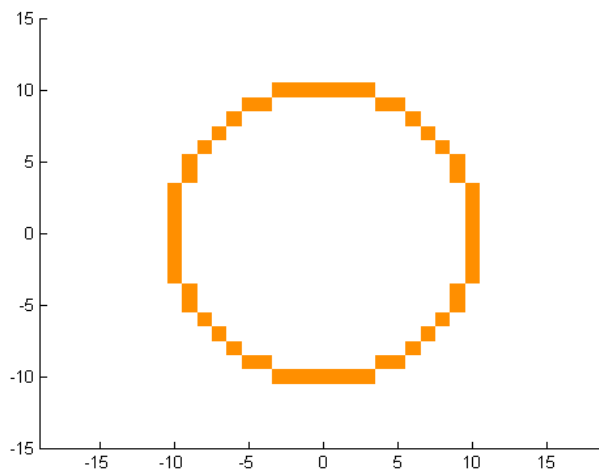
Σχεδιασμός Κύκλου και Έλλειψης

Η υλοποίηση των αντίστοιχων αλγορίθμων σε Matlab γίνεται από τις function **circle** και **ellipse**. Όπως μπορεί κανείς να παρατηρήσει οι συγκεκριμένες συναρτήσεις σχεδιάζουν μόνο τμήματα ενός κύκλου ή μιας έλλειψης. Αλλά η μετατροπή τους για να μπορούν να σχεδιάσουν πλήρως το σχήμα που θέλουμε είναι πολύ εύκολη και βασίζεται σε γνωστές συμμετρίες που έχουν ειπωθεί στην θεωρία.

Οπότε εμείς θα ασχοληθούμε λίγο με μια άλλη εντολή του Matlab. Την **axis equal**. Όπως μπορεί να δει κάποιος, αν σχεδιάσουμε ένα κύκλο με ακτίνα 10 το Matlab θα μας εμφανίσει στην οθόνη μας το ακόλουθο σχήμα.



Που πιο πολύ μοιάζει με έλλειψη παρά με κύκλο. Αν όμως εμείς στο τέλος του προγράμματός μας έχουμε γράψει την εντολή **axis equal** το ίδιο ακριβώς σχήμα θα μας εμφανίζεται



Που προφανώς μοιάζει πιο πολύ με κύκλο. Αυτό συμβαίνει γιατί με την εντολή αυτή λέμε στο πρόγραμμα να μας εμφανίζει τους δύο άξονες με την ίδια κλίμακα, κάτι που **δεν** γίνεται by default από το ίδιο το πρόγραμμα.

Οι αλγόριθμοι του κύκλου και της έλλειψης σχεδιάζουν σχήματα που έχουν σαν “κέντρο” το (0,0) αν εμείς όμως θέλουμε να παράγουμε σχήματα με κέντρο το (X_0, Y_0) δεν χρειάζεται να κάνουμε τίποτα περισσότερο από :

- 1) Να ζητήσουμε από τον χρήστη να έχει εισάγει το (X_0, Y_0) σαν είσοδο στην συνάρτηση.
- 2) Αντί να του πούμε να σχεδιάσει το (x, y) να του πούμε να σχεδιάσει το $(x+X_0, y+Y_0)$.