## Εισαγωγή

Η ανάλυση που ακολουθεί έχει ως σκοπό την παρουσίαση της υπολογιστικής λύσης της κλασσικής εξίσωσης Poisson. Για την επίλυση, έγινε χρήση της μεθόδου των πεπερασμένων διαφορών(Finite Diferences) και της τεχνικής Successive overrelaxation (SOR) για τον υπολογισμό της μήτρας δυναμικού.

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι Typescript.

Η επίλυση έγινε για δύο διαστάσεις, με την εξίσωση πηγής να είναι

## Θεωρία

### Εξίσωση Poisson

Σε μία, διάσταση, η εξίσωση έχει την μορφή :

όπου φ(χ) ορίζεται το ηλεκτρικό δυναμικό και S(x) η γραμμική πυκνότητα φορτίου.

Αντίστοιχα, σε τρείς διαστάσεις, η εξίσωση γίνεται:

όπου.

### Finite Differences Method (FDM)

Για τον υπολογισμό της μήτρας δυναμικού κάνουμε την παραδοχή ότι ο χώρος αποτελείται από ένα πλέγμα με ενιαία σημεία στα οποία η συνάρτηση δυναμικού μπορεί να εφαρμοστεί. Αν τα σημεία απέχουν μεταξύ τους απόσταση ***h,*** τα σημεία που ανήκουν στο πλέγμα είναι διαθέσιμα με τις παρακάτω εξισώσεις

Χρησιμοποιώντας το ανάπτυγμα Taylor και αντικαθιστώντας τα παραπάνω καταλήγουμε στην εξίσωση

Στην παραπάνω εξίσωση, κάνουμε χρήση της μεθόδου SOR, οπότε προσθέτουμε και τον παράγοντα

Η τελική μορφή γίνεται

## Επίλυση

Οι μεταβλητές που χρησιμοποιήθηκαν είναι:

Ω = 0,3

Επαναλήψεις = 3000

Μέγεθος πλέγματος = 30

### Βήμα 1 – Δήλωση Μεταβλητών

Η δήλωση των μεταβλητών αφορά :

1. Τον δισδιάστατο πίνακα φορτίου
2. Τον δισδιάστατο πίνακα δυναμικού
3. Την συνολική ενέργεια
4. Τον πίνακα με το ηλεκτρικό πεδίο του άξονα x
5. Τον πίνακα με το ηλεκτρικό πεδίο του άξονα y

### Βήμα 2 – Προετοιμασία πίνακα φορτίου

Μηδενίζουμε τον πίνακα φορτίου με τον ακόλουθο κώδικα:

  for (let i = 0; i < this.SIZE; i++) {

     for (let j = 0; j < this.SIZE; j++) {

        this.chargeMatrix[i][j] = 0;

         }

    }

### Βήμα 3 – Δημιουργία συνοριακών συνθηκών

Το πρώτο βήμα για να εφαρμόσουμε τις συνοριακές συνθήκες είναι να γνωρίζουμε ποια σημεία της μήτρας ανήκουν στα ακριανά σημεία. Για αυτό το λόγο, δημιουργούμε την ακόλουθη συνάρτηση:

 private isAtBoundaries(i: number, j: number): boolean {

    if (

      i == 0 ||

      j == 0 ||

      i == this.SIZE - 1 ||

      j == this.SIZE - 1 ||

      i < 0 ||

      j < 0 ||

      i > this.SIZE - 1 ||

      j > this.SIZE - 1

    ) {

      return true;

    }

    return false;

  }

Στην περίπτωση που ένα σημείο ανήκει σε άκρο, κατά τον υπολογισμό του δυναμικού αυτό το σημείο επιστρέφει 0, όπως φαίνεται και παρακάτω στην συνάρτηση υπολογισμού του δυναμικού.

### Βήμα 4 – Δημιουργία συνάρτησης πηγής

Η συνάρτηση πηγής σε μορφή κώδικα είναι :

  Math.pow(h, 2) \* ((cos(x \* Math.PI)) + Math.pow(y, 2) \* (sin(y \* Math.PI)))

### Βήμα 5 – Επαναληπτική διαδικασία βελτίωσης δυναμικού

Αρχικά υπολογίζουμε το φορτίο με τον ακόλουθο κώδικα

  private calculateCharge(i: number, j: number): number {

    const x = this.getRealXY(i);

    const y = this.getRealXY(j);

    const result = this.chargeEquation(x, y, this.H);

    return result;

  }

Όπου

  private getRealXY(i: number) {

    return i / (this.SIZE - 1);

  }

***SIZE*** είναι η μεταβλητή που περιέχει τον αριθμό των σημείων του πλέγματος.

Στη συνέχεια υπολογίζουμε με ***N*** επαναλήψεις το δυναμικό.

 for (let k = 0; k < this.ITERATIONS; k++) {

      for (let i = 0; i < this.SIZE; i++) {

        for (let j = 0; j < this.SIZE; j++) {

          const prevVoltageValue = this.calculatePotential(i, j);

          this.lab.voltageMatrix[i][j] = prevVoltageValue;

        }

      }

Όπου η calculatePotential είναι:

  private calculatePotential(i: number, j: number): number {

    if (this.isAtBoundaries(i, j)) return 0.0;

    let p = (1 - this.OMEGA) \* this.lab.voltageMatrix[i][j] +

      (this.OMEGA / 4.0) \*

      (this.lab.voltageMatrix[i + 1][j] +

        this.lab.voltageMatrix[i - 1][j] +

        this.lab.voltageMatrix[i][j + 1] +

        this.lab.voltageMatrix[i][j - 1] +

        this.chargeMatrix[i][j]);

    return p;

  }

### Βήμα 6 – Υπολογισμός Δυναμικού και Ενέργειας

### Βήμα 7 – Υπολογισμός Ηλεκτρικού πεδίου

Για τον υπολογισμό του ηλεκτρικού πεδίου χρειαζόμαστε τον πίνακα του δυναμικού. Για την διάσταση ***x,*** η συνάρτηση υπολογισμού είναι

   // initial calculation

    for (let i = 0; i < size - 1; i++) {

      if (!this.xVector[i]) {

        this.xVector[i] = [];

      }

      for (let j = 0; j < size - 1; j++) {

        this.xVector[i][j] =

          -(this.voltageMatrix[i + 1][j] - this.voltageMatrix[i][j]) / this.h; // eq 24

      }

    }

    // normalization

    for (let i = 0; i < size - 1; i++) {

      for (let j = 0; j < size - 1; j++) {

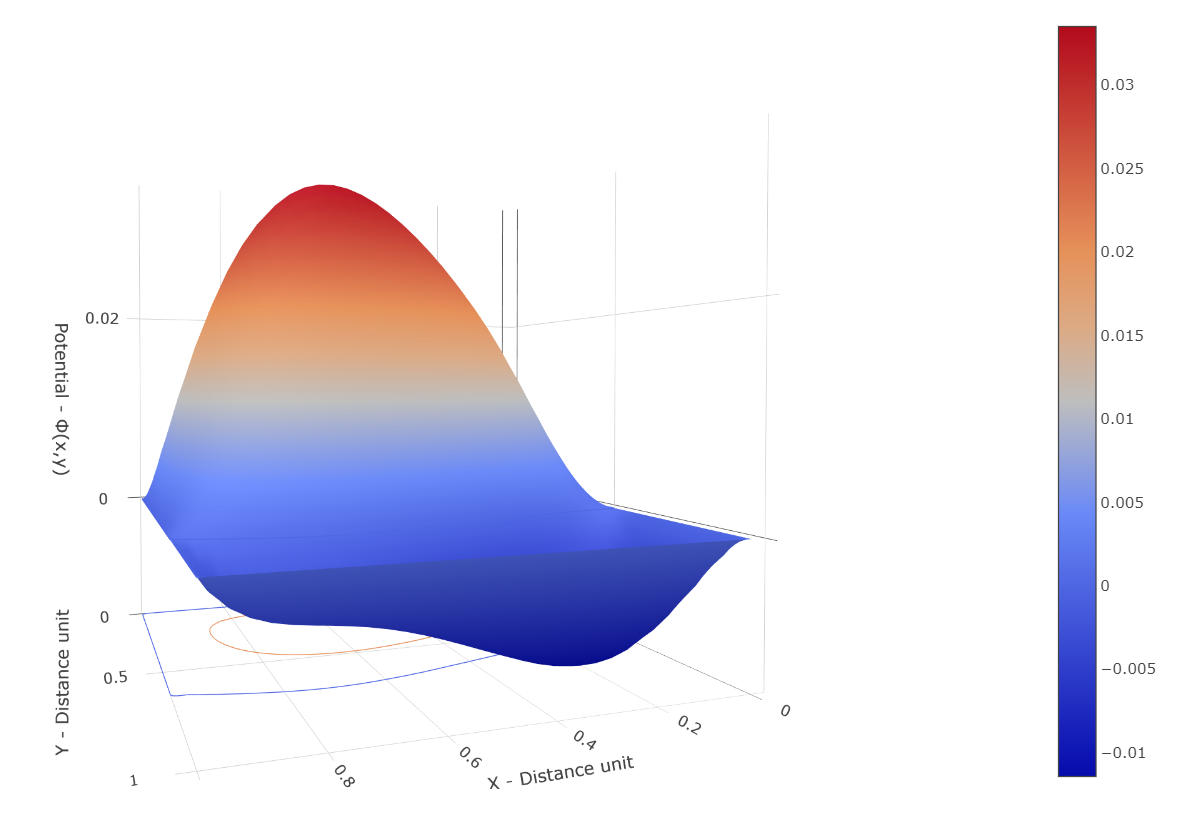
        this.xVector[i][j] =

          0.5 \* (this.xVector[i][j + 1] + this.xVector[i][j]); // eq 26

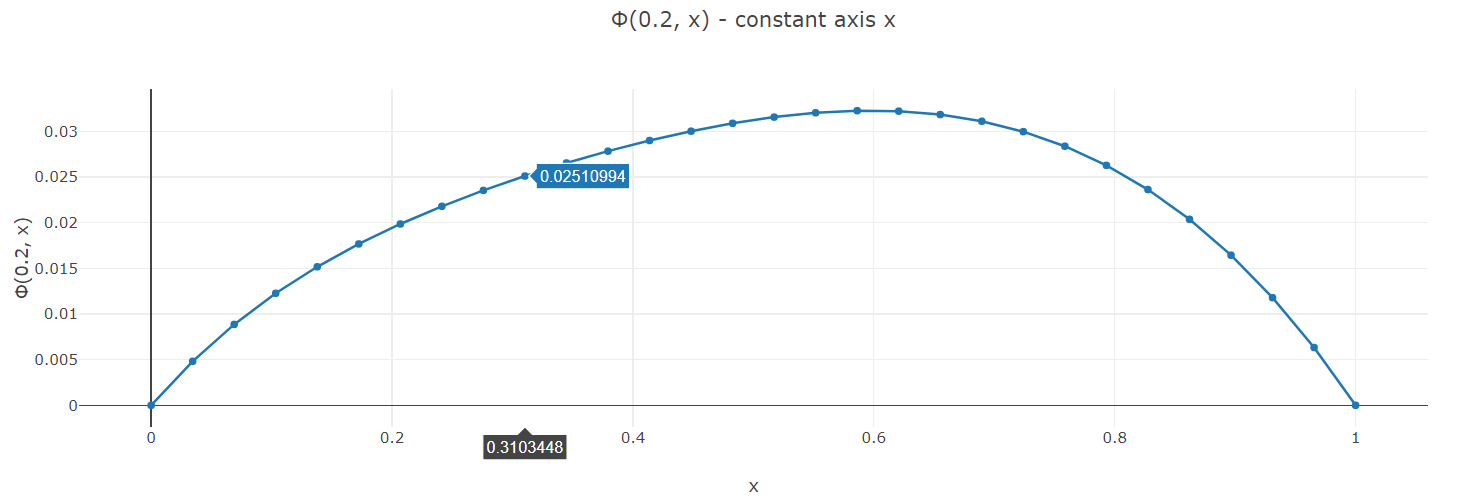
      }

    }

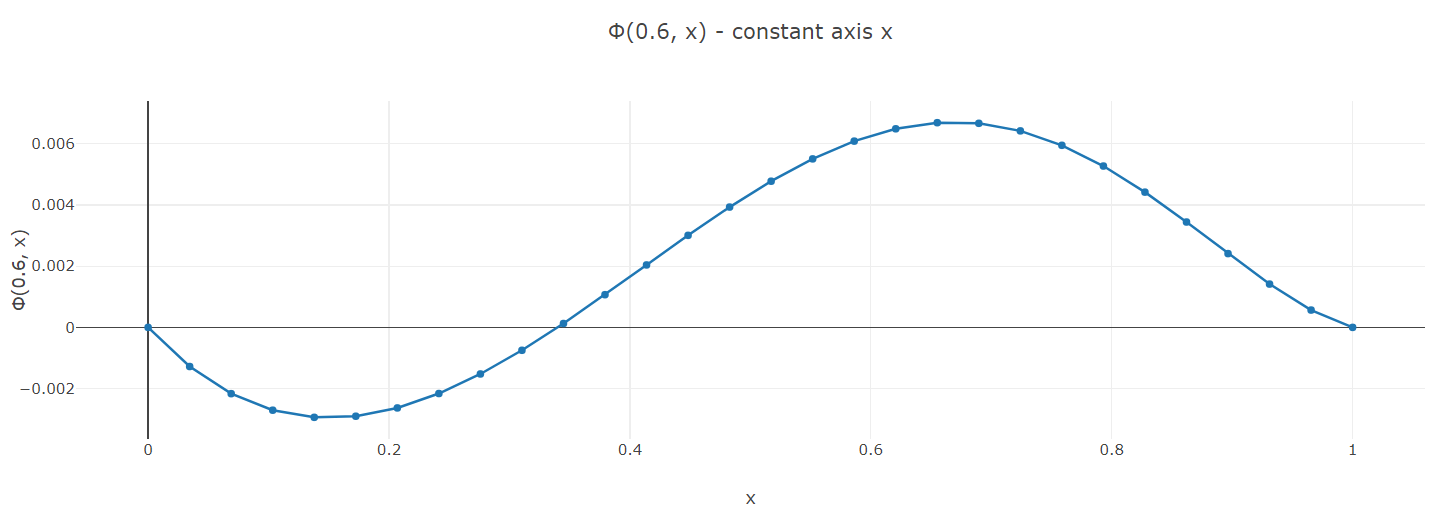
### Βήμα 8 – Γραφικές παραστάσεις



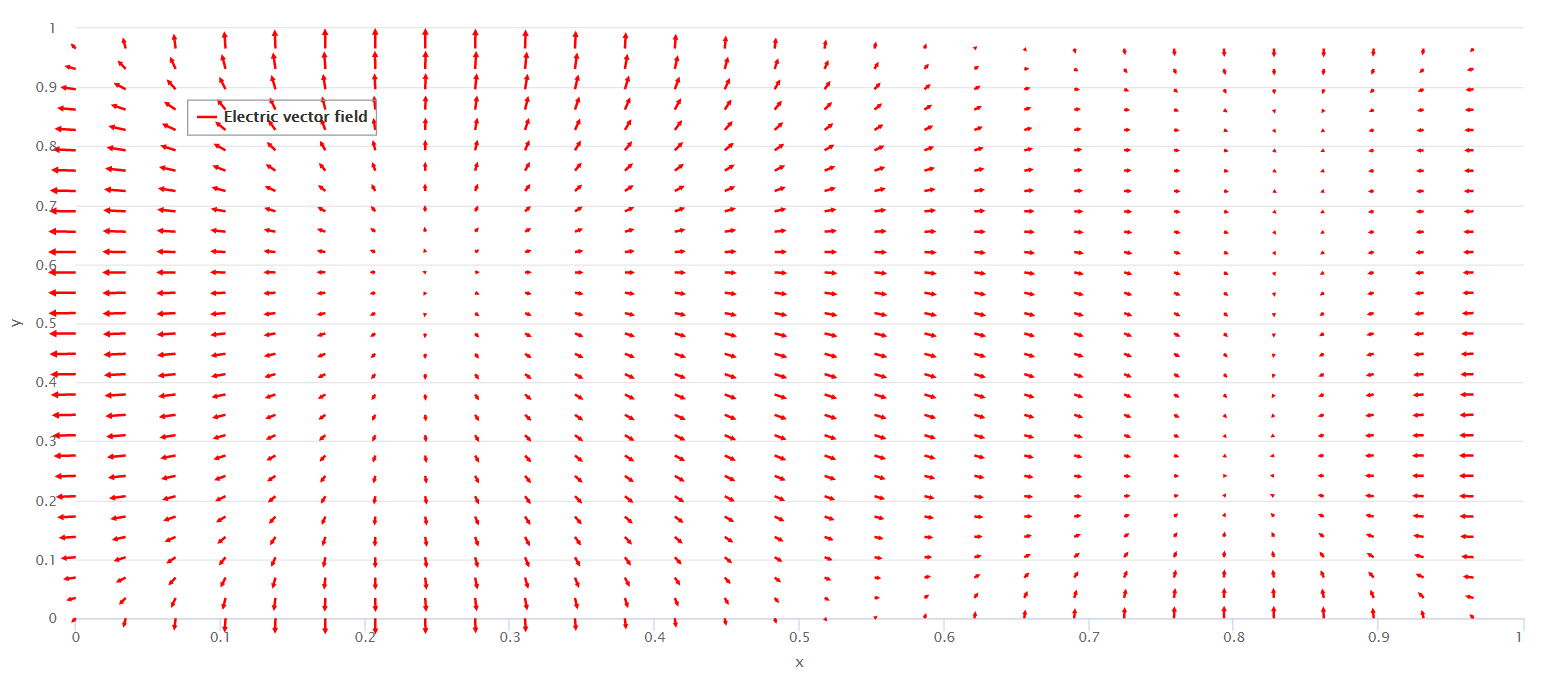
Εικόνα 1- Δυναμικό στις διαστάσεις x,y



Εικόνα 2 - Δυναμικό για x = 0,2 (Σημείο 6)



Εικόνα 3 - Δυναμικό για x = 0,6 (Σημείο 18)



Εικόνα 4 – Ηλεκτρικό πεδίο