

به نام خدا



تاریخ:

تعهدنامه اصالت اثر



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

گزارش کار پروژه

آموزش استفاده از زبان برنامه نویسی پایتون برای کاربردهای الکترو مغناطیس

نگارش

عرفان قصری

استاد راهنما

دکتر غلامرضا مرادی

دکتر احد توکلی

استاد مشاور

دکتر غلامرضا مرادی

بهار ۱۴۰۴

## چکیده

در این گزارش کار تلاش می‌رود تا گزارشی کامل از آنچه تا کنون درباره پروژه انجام داد شده را مستند کنم. به طور کلی در این پروژه تلاش می‌شود تا روشی ساده و کارآمد برای استفاده دانشجویان و اساتید از زبان پایتون به جهت حل مسائل الکترومغناطیسی ارائه شود؛ بدین جهت تلاش می‌شود تا در تمامی مراحل سادگی در استفاده، چشم نواز بودن نتایج و یکپارچگی روش‌ها اولویت قرار بگیرند و به موارد موارد زاید در پس زمینه رسیدگی شود.

تا کنون در این پروژه تلاش کردم تا از روش‌های مختلف به حل معادله پواسون بپردازم تا در نهایت به کمک اساتید راهنمای محترم یکی از این روش‌ها را برای ادامه پروژه برگزینیم؛ از این رو **چندتا** سیستم‌های متفاوت برای پیاده‌سازی این پروژه را در این گزارش بررسی خواهیم کرد و به مزایا و چالش‌های احتمالی هر یک می‌پردازیم.

## واژه‌های کلیدی:

گزارش کار، حل عددی، انیمیشن، الکترومغناطیس، آموزش

صفحه

## فهرست مطالب

چکیده.....	۱
فصل اول مقدمه (بررسی کلی پروژه) مقدمه.....	۱
۱-۲- بررسی اجمالی پایتون.....	۲
۱-۲- بررسی اجمالی پروژه.....	۳
فصل دوم حل عددی معادله پواسون.....	۶
فصل سوم آینده پروژه.....	۱۲
آینده پروژه.....	۱۳
فصل چهارم شیوه نصب شیوه نصب.....	۱۴
Abstract.....	۱۸

صفحه

## فهرست اشکال

شکل ۱	فضاهای برداری سه بعدی.....	۷
شکل ۲	نمایش دو بعدی فضای برداری.....	۸
شکل ۳	حل مسئله دی الکتریک در خازن مسطح(شکل سمت چپ توزیع بار، شکل وسط توضیح گذردهی و شکل چپ توزیع ولتاژ).....	۹
شکل ۴	توزیع شدت میدان الکتریکی در مثال خازن و دی الکتریک.....	۱۰
شکل ۵	فضای برداری میدان الکتریکی در مثال خازن و دی الکتریک.....	۱۱
شکل ۶	مثال نصب Numpy.....	۱۶

## فهرست علائم

### علائم یونانی

ثابت گذردهی خلا  $\epsilon$

## فصل اول

### مقدمه (بررسی کلی پروژه)

## مقدمه

در این بخش به بررسی پایتون و برخی از ابزارها و چالش‌های پروژه میپردازیم.

### ۲-۱- بررسی اجمالی پایتون

پایتون یکی از محبوب‌ترین زبان‌های برنامه‌نویسی در جهان است که به دلایلی از جمله:

۱. **سادگی و خوانایی:** نحو<sup>۱</sup> ساده و قابل فهم پایتون، یادگیری آن را برای افراد مبتدی آسان‌تر می‌کند و باعث افزایش سرعت توسعه نرم‌افزارها و پایگاه‌کدها می‌شود.

۲. **کتابخانه‌های (ماژول‌های) گسترده:** پایتون دارای مجموعه‌ای وسیع از کتابخانه‌ها و ابزارهای تخصصی مانند NumPy، Pandas، Matplotlib و SciPy است که در زمینه‌های مختلفی از جمله زمینه‌های محاسباتی فعال هستند و کار توسعه را به مراتب ساده‌تر میکنند.

۳. **جامعه فعال:** پشتیبانی گسترده، پیشرفت دائمی و مستندات فراوان باعث شده تا حل مشکلات و یادگیری این زبان ساده‌تر باشد.

البته استفاده از این زبان نیز بدون چالش نیست و در کاربرد مد نظر ما نیز به برخی از این چالش‌ها برخورد میکنیم، از جمله این چالش‌ها میتوان به موارد زیر اشاره کرد:

۱. **کندی اجرا:** پایتون نسبت به زبان‌هایی مثل C++ یا Java کندتر اجرا می‌شود، زیرا تفسیر است و بهینه‌سازی سطح پایین ندارد.

۲. **وابستگی به کتابخانه‌های خارجی:** بسیاری از قابلیت‌های پایتون به کتابخانه‌های جانبی متکی است، که ممکن است باعث ناسازگاری نسخه‌ها شود.

دیده میشود که ویژگی مانند تفسیر بودن این زبان اگرچه باعث سادگی استفاده از این زبان میشود اما در جنبه‌ای دیگر باعث کند شدن این زبان نسبت به دیگر زبان‌های مطرح میشود.

---

<sup>1</sup> Syntax

همچنین وجود کتابخانه های گسترده شخص ثالث<sup>۲</sup> که دائما در حال بروزرسانی و اضافه کردن قابلیت های جدید به این زبان هستند اما ترکیب این موضوع با مفسری بودن زبان، به ناسازگاری های مختلفی می انجامد (برای مثال تفاوت عملکرد کد یکسان برای افراد متفاوت به دلیل تفاوت در وابستگی های<sup>۳</sup> کد به کتابخانه های شخص ثالث، سیستم عامل، اسامی متغیر ها و ... یا تغییر عملکرد ویا از کار افتادن کلی کد در طول زمان به دلیل تغییر کتاب خانه ها که از این رو پایتون زبانی بدنام<sup>۴</sup> است) البته این چالش ها بدون راه حل نیست و استفاده از زبان های سریع تر همانند C++ برای انجام محاسبات سنگین (همانند آنچه در NumPy انجام میشود) بسیار معمول است که باعث افزایش سرعت در عین حفظ سادگی میشود؛ از این رو استفاده از کتابخانه ها پایتون عموما نسبت به پیاده سازی مستقیم توابع با محاسبات بالا ارجحیت دارند.

برای حل مشکلات مربوط به کتابخانه های خارجی نیز ابزار های بسیار متنوعی از جمله containerization<sup>۵</sup> همانند Docker<sup>۶</sup> وجود دارند.

البته این دو مورد تنها مشکلات پایتون نبودند و پایتون مشکلات دیگری همچون: مصرف بالای حافظه، GIL (Global Interpreter Lock) را نیز دارد.

## ۲-۱- بررسی اجمالی پروژه

از آنجا که تا کنون تنها به حل معادله پواسون پرداخته شده است برخی از گزاره ها صحیح نبوده یا با سو گیری باشند.

<sup>۲</sup> Third party

<sup>۳</sup> Dependency

<sup>۴</sup> پایتون به داشتن Dependency Hell مشهور است.

<sup>۵</sup> کانتینری کردن

<sup>۶</sup> داکر



- در این پروژه قصد داریم تا یک (یا چند) ماژول پایتونی بسازیم تا ۲ مورد عمده را برآورده کند:
۱. اساتید بتوانند با آن به سادگی محتوای چندرسانه ای به جهت تسهیل فهم مطالب و دادن دید از درسی انتزاعی الکترومغناطیس تولید کنند.
  ۲. دانشجویان بتوانند به راحتی با استفاده از آن مثال هایی را در شرایط مختلف (احتمالا پیچیده تر از آنچه به طور تحلیلی قابل حل است) از مسائل الکترومغناطیسی را حل کنند.
- پس به سادگی از آنچه گفته شد برداشت میشود که ویژگی های زیر در این پروژه حیاتی اند:
۱. سادگی در استفاده: استفاده پروژه باید به قدری ساده باشد که افرادی که تا کنون از پایتون استفاده نکرده اند نیز بتوانند با حداقل تلاش از پروژه (دست کم در حد مثال ها) استفاده کنند و استفاده از آن نباید منوط به دانش قبلی پایتون و یا کتابخانه های شخص ثالث (هرچند ساده) داشته باشد.
  ۲. سادگی در راه اندازی اولیه: اولین استفاده از پروژه با توجه به نیاز پروژه به نصب پیش نیازهای آن به تبع پیچیده تر از استفاده های بعدی ست اما این تفاوت نباید چشمگیر باشد، البته با توجه به بده بستانی که میان مزایا برخی تکنولوژی ها (همچون Docker و Manim) این مرز باید با کمک اساتید راهنما ترسیم شود.
  ۳. سادگی در یادگیری و داشتن داکيومنتیشن قوی
  ۴. قابل فهم و چشم نوازی نتیجه که عمدتا از نوع تصویری یا اینمیشنی خواهد بود
  ۵. مثال های زیاد استفاده به جهت داشتن الگو برای استفاده
  ۶. غنای زیاد: الکترومغناطیس سرشار از مسائل متنوع است و این پروژه باید توانایی تسهیل حل سهم خوبی آنها را داشته باشد
- همچنین موارد همچون توسعه پذیری و سرعت عملکرد و یکپارچگی با دیگر کتابخانه ها نیز از موارد مهم در این پروژه هستند.



## فصل دوم

### حل عددی معادله پواسون

تا کنون بر روی حل معادله پواسون در دو بعد به عنوان پایه ای برای پروژه ام استفاده کردم با حل این معادله به کلیت مسئله دید بهتری پیدا خواهیم کرد و راحت تر میتوان میان برخی بده بستان ها انتخاب کرد.

به طور کلی حل این معادله را میتوان به سه قسمت تقسیم کرد:

#### ۱. طراحی مسئله

باید بتوان توزیع های متنوع از بار، محیط های گوناگون و شرایط مرزی قابل تعریف باشند. در این قسمت چالش چندانی وجود ندارد و مسئله تنها پیاده سازی شرایط متنوع است، برخی از این اشکال در زیر مشخص اند

کمان، دایره، بیضی، خط، دو خط موازی (عموما این معادله در یافتن ظرفیت خازنی کاربرد دارد از این روی این شکل مد نظر است)، مستطیل، شش ضلعی، چند ضلعی دلخواه

#### ۲. حل عددی مسئله

حل کننده مسئله باید سرعت بالایی داشته باشد پس بهتر است از کتابخانه های پایتون استفاده کنیم البته پیاده سازی الگوریتم های حل نیز بی بهره نیستند و امکان توسعه پذیری بیشتری به ما میدهند.

به تبع به دلیل بده بستان بین سرعت و کاربرد در اینجا باید نظر اساتید راهنما را جویا شد.

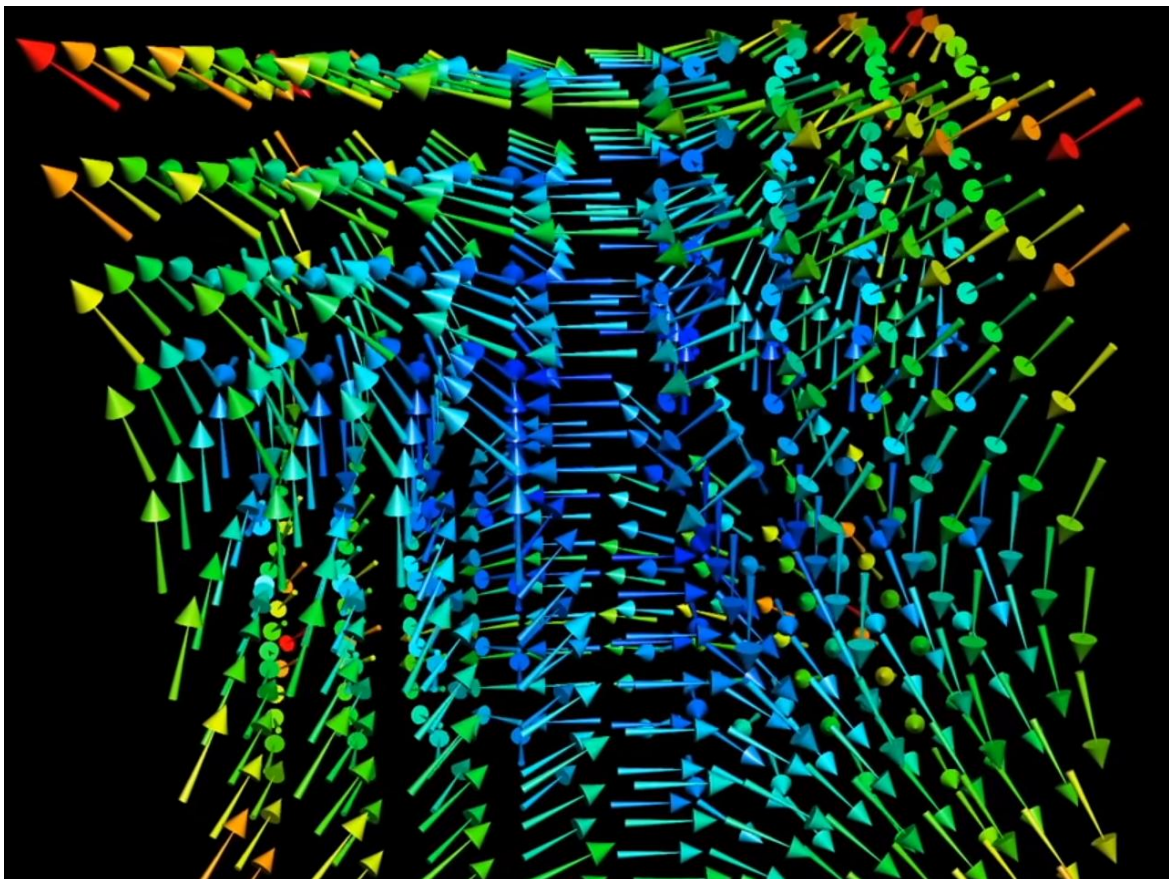
مقایسه بین `finite solver.py` (زمان حل در حدود ۱۰ ثانیه برای شبکه ۱۰۰ در ۱۰۰) و مثال های نوت بوک `examples` (زمان حل + نمایش نتایج ۰.۷ ثانیه ای برای شبکه ۲۵۶ در ۲۵۶ در بلوک شماره ۶) گواه خوبی از این موضوع است.

همچنین بدیهی است که استفاده از حل کننده های داخلی این مسائل باعث کمتر شدن بار پیاده سازی پروژه شده و وقت بیشتری برای سایر جنبه های پروژه باقی میگذارند.

### ۳. نمایش نتایج

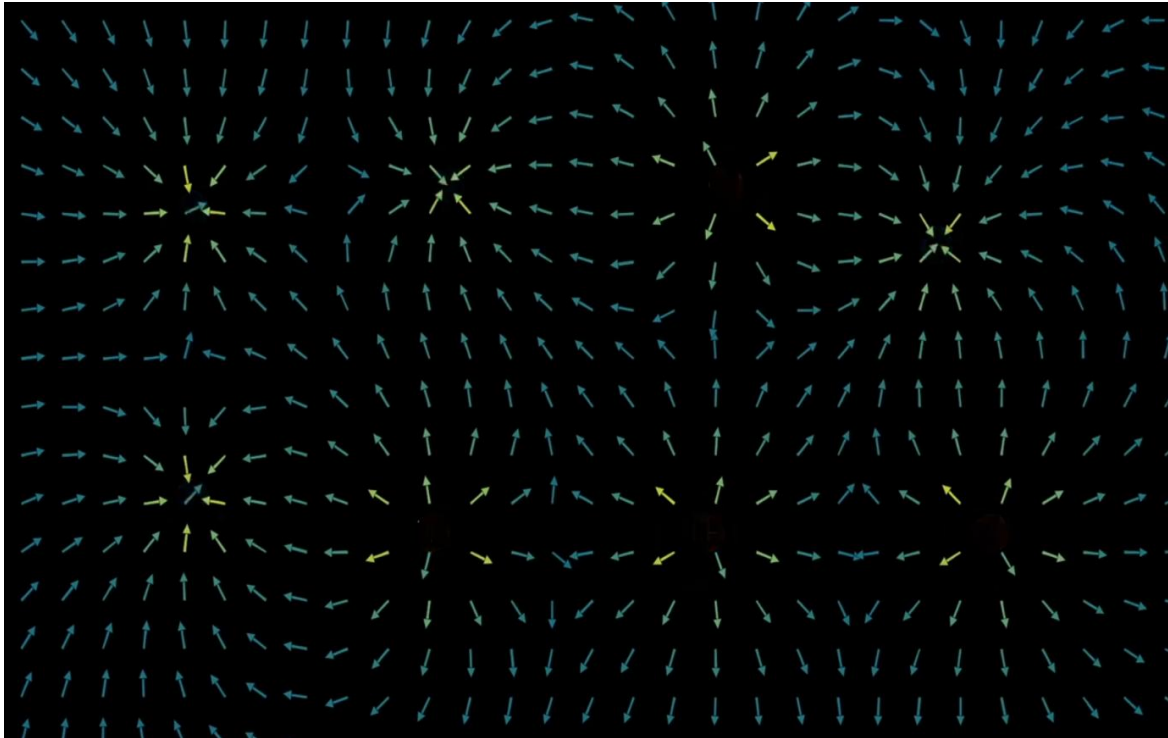
نتایج به شیوه های مختلفی قابل نمایش اند اما عموماً فرم پوسته سه بعدی برای ولتاژ و فضای برداری برای میدان الکتریکی دارند، در فضای دو بعدی عمدتاً نمایش دامنه با رنگ به دلیل سهولت اولویت دارد اما نمایش با اشکال سه بعدی نیز امکان پذیر است.

چالش این قسمت فضای سه بعدی است، در فضا سه بعدی به سرعت فضا بیش از حد پیچیده شده و چیز زیادی قابل درک نیست.



شکل ۱ فضا های برداری سه بعدی

میبینیم که علاوه بر نرمالایز کردن اندازه بردار ها نمایش آنها در رنگشان باز هم نتایج نسبتاً نامفهوم اند.



شکل ۲ نمایش دو بعدی فضای برداری

البته در فضای دو بعدی نیز باید توجه داشت که نتایج نباید بیش از حد شلوغ شوند پس حتی اگر تعداد نقاط زیاد بود (حل با رزولوشن بالایی انجام گرفت) نیز بهتر است در نهایت مقداری از نتایج را جدا کنیم یا under sample کنیم تا نتایج بهتر دیده شوند.

نمونه ای از این فرایند سه مرحله ای را در مثال دی الکتریک بین یک خازن مسطح میبینیم:

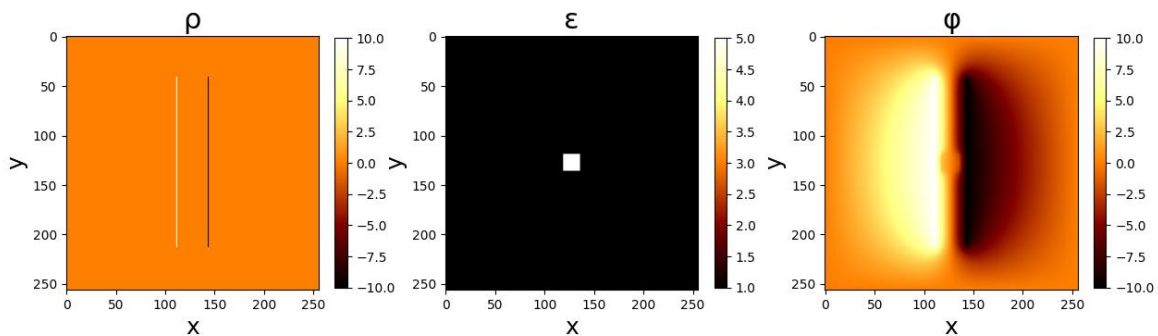
```
N = 256
capacitor = ops.plate_capacitor(center=(N/2-1, N/2-1), length=N/1.5,
distance=N/8, rotation=90, values=[-10, 10])
U = capacitor(np.zeros((N, N)))

Eps = np.ones((N, N))
Eps = ops.rectangle(Eps, center=(N/2-1, N/2-1), wh=(N//16, N//16),
filled=True, value=5)

phi = solve_poisson(U=U, Eps=Eps)
plot_rho_eps_phi(rho=U, eps=Eps, phi=phi)
```

در خط اول شبکه را ۲۵۶ تایی تعریف میکنیم؛ سپس با استفاده از ترسیم شکل که توضیح داده شده، به ترسیم یک خازن در مرکز شبکه به طول ۶۶ درصد شبکه و با فاصله بین صفحات ۱۲.۵ درصد شبکه‌ای ترسیم میکنیم همچنین در یک صفحه بار ۱۰ واحدی و در دیگری بار منفی ۱۰ واحدی قرار میدهم (در این مثال  $\epsilon$  برای فضای آزاد ۱ در نظر گرفته شده است پس مقادیر اساساً بیانگر واحد SI نیستند) در نهایت خازن مد نظر را بر روی فضای بار های رسم میکنیم و همین عمل را برای هدایت الکتریکی نیز تکرار میکنیم.

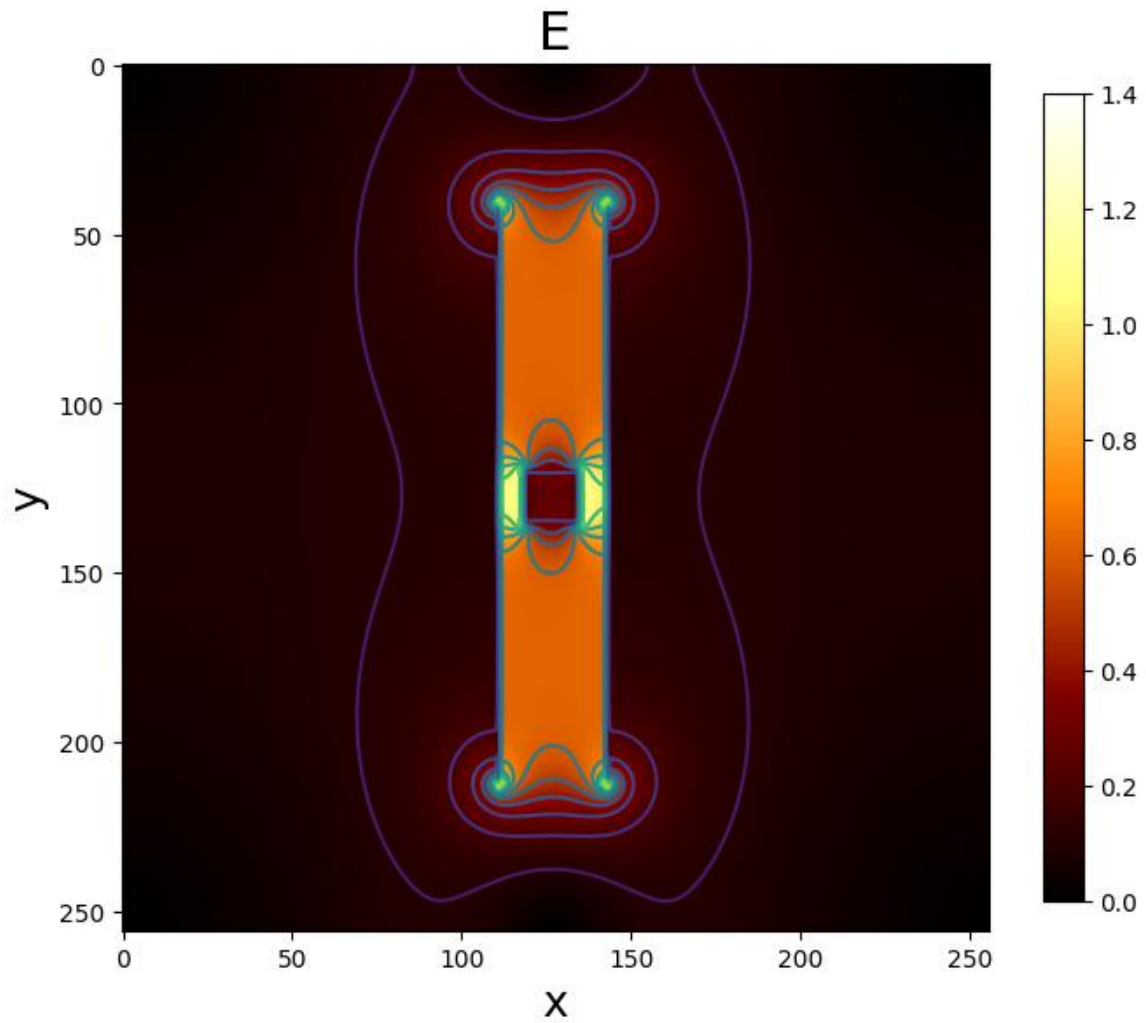
حال با استفاده از حل کننده عددی مقادیر مرتبط با ولتاژ استخراج میشوند و اساساً مسئله حل میشود. در نهایت برای نمایش نتایج نیز از توابع مخصوصشان استفاده میکنیم (این تابع به خصوص چندان مناسب پیاده سازی نشده اما میتوان آنرا بهتر کرد به طوری که با هر تعداد ارگيومنت ورودی به خوبی کار کند)



شکل ۳ حل مسئله دی الکتریک در خازن مسطح (شکل سمت چپ توزیع بار، شکل وسط توضیح گذردهی و شکل چپ توزیع ولتاژ)

حال از توزیع ولتاژ، میدان الکتریکی را نیز استخراج میکنیم:

```
plot_E_intensity(phi=phi)
```

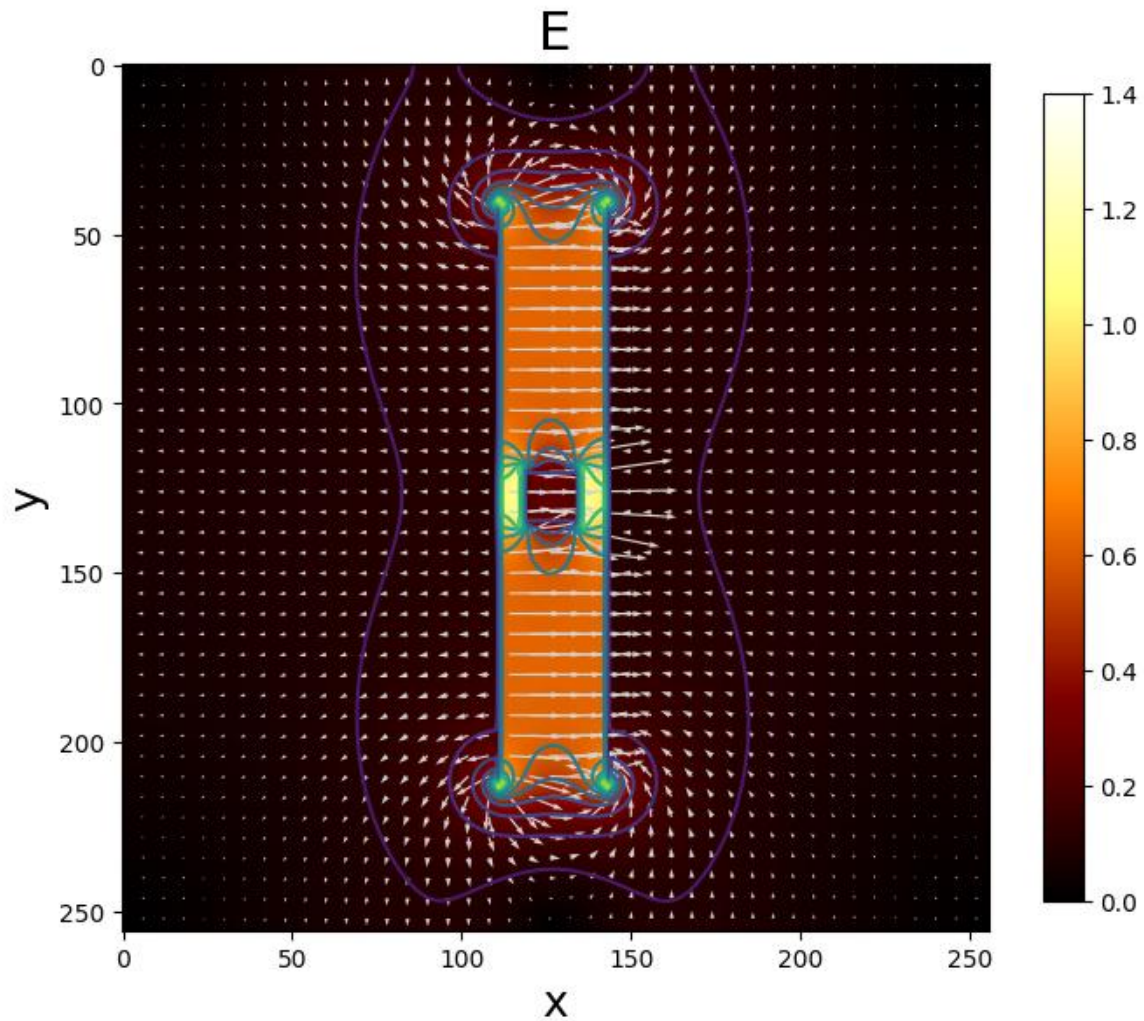


شکل ۴ توزیع شدت میدان الکتریکی در مثال خازن و دی الکتریک

البته به سادگی میتوانیم فضای برداری میدان را نیز به این نمودار اضافه کنیم:

```
plot_E_intensity(phi=phi, field_vector=True)
```





شکل ۵ فضای برداری میدان الکتریکی در مثال خازن و دی الکتریک

طبیعتاً این مثال و توابع مربوط به آن کامل نیستند و مواردی همچون توزیع انرژی الکتریکی، یکپارچگی برخی توابع و قابلیت تولید انیمیشن از این فضا یا فضا در حال تغییر از اهداف کوتاه مدت برای اضافه شدن به پروژه هستند.



## فصل سوم آینده پروژه

## آینده پروژه

در این قسمت به برخی از قابلیت هایی که قصد دارم به پروژه اضافه کنم به شکل مختصر و همچنین کارموردی که باید تا پایان پروژه صورت بگیرد به ترتیب اولویت افزوده شدن میپردازیم:

۱. تهیه لیست اولیه از حداقل مسائلی که باید توسط پروژه قابل حل باشند همانند معادله لاپلاس، معادله امپر و غیره.

۲. افزودن قابلیت پردازش و نمایش درخور فضای سه بعدی

۳. قابلیت ایجاد انیمیشن(احتمالا با ایجاد توابعی که به عنوان رابط بین پروژه و کتابخانه Manim عمل کنند)

۴. تکمیل لیست شکل های قابل پیاده سازی

۵. افزودن دیگر مسائل و توابع و متد های مرتبط با آنها

۶. یکپارچه سازی یا تجزیه عملکرد پروژه: بسته به اینکه پروژه بهتر از ابزار خاص برای هر مسئله داشته باشد یا اینکه بهتر است ابزاری باشد برای حل هر مسئله ای.

۷. ساخت مثال های زیاد توسط پروژه که در حکم تست و دیباگ پروژه نیز عمل میکند.

۸. مستند سازی پروژه از جمله شیوه نصب و استفاده

۹. تهیه و نگارش پایان نامه

## فصل چهارم شیوه نصب

## شیوه نصب

در ابتدا دو مورد لازم به ذکر میباشند:

۱. در طول چند سال اخیر برای مقاصد بسیار متنوعی از پایتون استفاده کردم بسیار از کتاب خانه ها از پیش روی سیستم من موجود بودند پس ممکن است برخی از وابستگی ها از قلم بیفتند(در نهایت پروژه، این مرحله را روی چند سیستم مختلف تست خواهم کرد پس این مشکل در نسخه نهایی پروژه وجود نخواهد داشت.
۲. این تنها یک روش از استفاده از پروژه است و روش های دیگر و شاید از برخی نقطه نظر ها بهتری نیز وجود دارند.

برای نصب و اجرای آنچه در این گزارش آمده دستکم به موارد زیر نیاز است:

- نسخه ای پایتون (پروژه تا کنون روی پایتون ۳.۱۱.۲ نوشته شده است)
  - یک ویرایشگر کد<sup>۱</sup> یا IDE<sup>۲</sup> مربوط به پایتون ([VS Code](#) پیشنهاد میشود)
  - Jupiter notebook
- در صورتی که از VS code استفاده شود این افزونه ها نیز نصب شوند:
- Python extension pack (در هنگام نصب گزینه add python to path زده شود)
  - Jupiter
- کتاب خانه های زیر نیز لازم است برای پایتون نصب شوند:
- Numpy
  - Matplotlib
  - Jupyterlab

<sup>۱</sup> Code editor

<sup>۲</sup> Integrated Development Environment

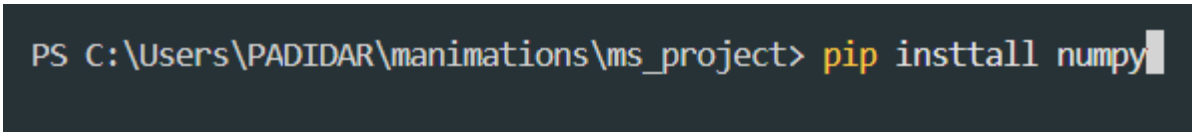
• Scipy

• Sys

برای نصب این کتاب خانه ها میتوان از پکیج منیجر پایتون با ترکیب زیر استفاده کرد:

`pip install muduleName`

برای مثال برای نصب numpy کافیست در ترمینال VS Code یا هر ترمینال دیگری وارد کنیم:



```
PS C:\Users\PADIDAR\animations\ms_project> pip install numpy
```

شکل ۶ مثال نصب Numpy

برای نصب اصل پروژه نیز میتوانید آنرا از:

<https://github.com/derfang/bachelors-project/archive/refs/heads/main.zip>

دانلود کرده یا اگر git نصب دارید با دستور:

`git clone https://github.com/derfang/bachelors-project.git`

یک نسخه از پروژه را تهیه کنید.

[Download Python | Python.org](#)

(پیوست ها صرفا به دلیل به هم نریختن فایل ورد خالی نیست)

## Abstract

In this report, I aim to document a comprehensive account of the progress made on the project so far. Broadly speaking, the project seeks to provide a simple and efficient method for students and professors to use Python for solving electromagnetic problems. thus, simplicity of use, visually appealing results, and methodological consistency are prioritized at every possible occurrence, while unnecessary complexities are handled in the background.

So far, I have explored various methods for solving the Poisson equation in this project, with the goal of selecting one approach—in consultation with the esteemed advising professors—to proceed with. Accordingly, this report will examine several different methods for implementing the project, discussing the advantages and potential challenges of each.

**Key Words: Numerical Solution, Animation, Electromagnetics, Education/Instruction.**



**Amirkabir University of Technology  
(Tehran Polytechnic)**

**Department of electrical engineering**

**Report of progress**

# **Tutorial for using python for electromagnetic applications**

**By  
Erfan Ghasry**

**Supervisor  
Dr.Gholamreza Moradi  
Dr. Ahad tavakoli**

**Advisor  
Dr.Gholamreza Moradi**

**spring of 2025**