به نام خدا

پروژه درس تئوری مدار

سید حسام الدین حسینی 400101034

نحوه ورودی گرفتن:

با توجه به این که مقاومت و سلف وخازن و منابع مستقل نیازی به بخش <dependence> ندارد پس تنها 5 ورودی میگیریم

<Type><name><node1><node2><value>

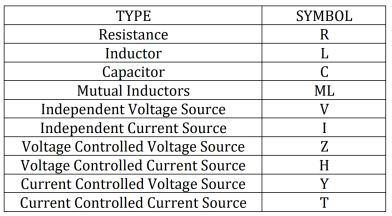
برای منابع وابسته داریم:

<<Type><name><node1><node2><value><dependent node1><dependent node2

برای سلف تزویج خواهیم داشت:

<Type><name1><node1\_L1>,< node2\_L1><value1><name2><node1\_L2><node2\_L2> <mutual Inductance>

و در مورد تایپ عناصر از همان جدول پیشنهادی استفاده کرده ایم



همچنین در نظر داریم زمانی که ML را به عنوان یه خط از ورودی میدهیم درواقع دو عنصر جدیدی اضافه میکنیم که شامل دو سلف وضریب تزویجشان است

توضیح کد:

در ابتدا فایل ورودی که از قبل نوشته ایم و تمام عناصر مدار در ان به فرمت هایی که بالا تر توضیح داده شد را لود میکنیم و همچنین مقدار سورس خود از command Window متلب دریافت میکنی ( برای سادگی sin(t) وارد میکنی)م سپس نود هایی که میخوایم ولتاژ بین آن دو نقطه را داشته باشیم وارد میکنیم



ما نیاز داریم که تعداد خط های فایل را بدانیم که برای این کار از دستور readlines استفاده میکنیم که ماتریس خط های رو بهمون میده و length این ماتریس همان تعداد خط ها است و سپس هر کدام از این خط ها را به تابعی میدهیم که بسته به اینکه فرمت ورودی کدام است المان ها ذخیره میکند



برای سلف و خازن مقدار المان را در اس ضرب میکنیم تا به حوضه لاپلاس ببریم در ادامه بسته به اینکه به ادمیتانس یا امپدانس المان نیاز داریم از خود مقدار ولیو یا عکس ولیو استفاده میکنیم

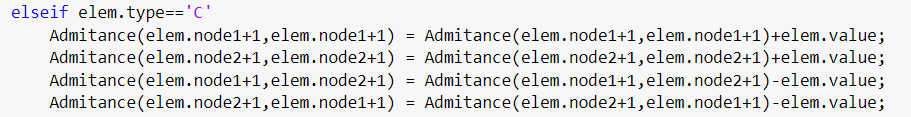
البته ما به این تابع مقدار سورس که گرفته ایم را هم پاس میدهیم تا وقتی منابع را لود میکند اگر مقدار آن منبع مشخص نشده بود یعنی آن همان منبع سورس ما است و مقدار آن را با مقدار ورودی جایگذاری میکنیم البته چون ورودی از نوع string است با تابع str2sym آن را به یک متغیر سیمبولیک تبدیلش میکنیم تا بتوان با آن کار کرد و بقیه مقادیر تنها عدد هستند پس به صورت عدد ذخیره شان میکنیم

ذخیره سازی المان ها هم به صورت کلاسی است که در فایل جداگانه نشوته شده است و شامل پراپرتی های مثل نود ها و اسم و تایپ سایر مقادر ورودی است و البته مقادیری دارند که مقدایر دهی نشده اند و در ادامه با حل مدار آن هارا مقدار دهی میکنیم

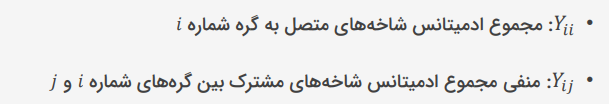
همچنین برای حل مدار نیاز داریم بزگترین نود را پیدا کنیم تا بهینه ترین الگوریتم را داشته باشیم این کار را با تابع های هایی که نوشته شده است هندل میکنیم و مقدارش را به تابع solvC پاس میدهیم تا مدار به صورت سیستمی با تجزیه و تحلیل نود حل کن را حل کند

که کارکرد این تابع اینگونه است که ابتدا با فور زدن روی تمامی المان ها ماتریس های لازم را میسازد

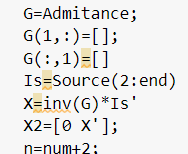
به طور مثال برای خازن داریم:



که ماتریس ادیمتانس را اینگونه میسازیم که المان های روی قطر اصلی از جمع کل ادیمتانس هایی که به نود مورد نظر وصل ان بدست می اید و ادمیتانس های بین اینکس i,j برابر است با منفی ادیمتانس مشترک

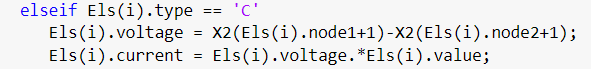


سپس با عملات های ماتریسی مدار را حل مکیند که ماتریس ایکس همان ماتریس ولتاز نود های مدار است



و در نهایت دوباره روی تمام المان ها فور میزد و مقادیر جریان و ولتاژ هر کدام از المان ها را مقدار دهی میکند

به طور مثال برای خازن داریم:



که مقدار ولتاژ را از هم ماتریس ولتاژ های نود ها با دادن نود آن ماتریس بدست میاوردیم و مقدار جریانش برابر با ضرب مقدار ولتاژ در مقدار ادیمتانسش است

در نهایت تابع فانلایز راداریم که مقادیر جریان و ولتاژ تمامی المان ها را در فایل خروجی میدهد و با توجه به دو نود داده شده در ورودی مقادیر ولتاژ بین آن دو نقطه و تابع تبدیل در حوضه لاپلاس را خروجی میدهد (تمامی مراحل حل در حوضه لاپلاس بود) با توجه به اینکه هنگامی که ما از ML استفاده میکنیم در واقع دو المان را با هم مقدار دهی میکنیم هر کجا ML داشتیم باید دو خروجی بدهیم که هر کدام از خروجی ها یکی از دو سلف تزویج هستند

وهمچنین با لاپلاس وارون گرفتن این مقادر را که خروجی گرفته در حوضه زمان با دستور fplotرسم میکند

