

ای نسخی نامه‌ای که تویی
ای آنسه‌ی حال شاهی که تویی
بیرون ز تونیست آنچه د عالم هست
از خود بطلب هر آنچه خواهی که تویی



دانشگاه صنعتی خواجه نصیرالدین طوسی

۱۳۰۷

دروشیهای عددی و تحلیلی

پرای مهندسین پرق پا متلپ

نویسندهان:

William Bober & Andrew Stevens

ترجمه

دکتر محمد صادق ابریشمیان

استاد دانشگاه صنعتی خواجه نصیرالدین طوسی

فهرست مطالب

| | | |
|----|---------------------------------------|------|
| ۱ | روشهای عددی برای مهندسین برق | |
| ۷ | مقدمه | ۱.۱ |
| ۷ | اهداف مهندسی | ۲.۱ |
| ۸ | برنامه‌نویسی حل عددی | ۳.۱ |
| ۹ | چرا متلب؟ | ۴.۱ |
| ۹ | زبان برنامه نویسی متلب | ۵.۱ |
| ۱۰ | قراردادهای این کتاب | ۶.۱ |
| ۱۲ | برنامه‌های نمونه | ۷.۱ |
| ۱۳ | مبانی متلب (MATLAB) | |
| ۱۳ | مقدمه | ۱.۲ |
| ۱۴ | پنجره‌های متلب | ۲.۲ |
| ۱۶ | نوشتن برنامه در متلب | ۳.۲ |
| ۱۸ | مبانی متلب | ۴.۲ |
| ۲۹ | ورودی و خروجی متلب | ۵.۲ |
| ۳۳ | روندهای برنامه متلب | ۶.۲ |
| ۴۲ | فایل‌های توابع متلب | ۷.۲ |
| ۴۷ | توابع بی‌نام | ۸.۲ |
| ۴۷ | گرافیک متلب | ۹.۲ |
| ۵۶ | کار با ماتریس‌ها | ۱۰.۲ |
| ۵۹ | کار با تابع یک بردار | ۱۱.۲ |
| ۶۰ | مثال‌های اضافی از کاراکترها و رشته‌ها | ۱۲.۲ |
| ۶۵ | درون‌بایی و تابع interp1 متلب | ۱۳.۲ |
| ۶۷ | تابع textscan متلب | ۱۴.۲ |
| ۶۹ | انتقال داده متلب به اکسل | ۱۵.۲ |
| ۶۹ | اشکال‌زدایی یک برنامه | ۱۶.۲ |
| ۷۲ | مدار RLC موازی | ۱۷.۲ |
| ۷۴ | مسائل | |
| ۷۴ | پروژه‌ها | |
| ۸۷ | ماتریس‌ها | |
| ۸۷ | مقدمه | ۱.۳ |
| ۸۷ | عملیات ماتریسی | ۲.۳ |
| ۹۱ | دستگاه معادلات خطی | ۳.۳ |

| | | |
|---|--|-------|
| ۹۵ | حذف گوسی | ۴.۳ |
| ۹۹ | روش گوس-جردن | ۵.۳ |
| ۱۰۰ | تعداد جوابها | ۶.۳ |
| ۱۰۰ | ماتریس معکوس | ۷.۳ |
| ۱۰۳ | مسئله مقدار ویژه | ۸.۳ |
| ۱۰۷ | مسائل | ۹.۳ |
| ۱۰۸ | پروژه‌ها | |
| ۴ ریشه‌های جبری و معادلات متعالی | | |
| ۱۱۳ | مقدمه | ۱.۴ |
| ۱۱۳ | روش جستجو | ۲.۴ |
| ۱۱۴ | روش دوبخشی | ۳.۴ |
| ۱۱۵ | روش نیوتون-رافسون | ۴.۴ |
| ۱۱۶ | تابع <code>fzero</code> و <code>roots</code> در متلب | ۵.۴ |
| ۱۱۶ | تابع <code>fzero</code> | ۱.۵.۴ |
| ۱۲۱ | تابع <code>roots</code> | ۲.۵.۴ |
| ۱۲۲ | پروژه‌ها | |
| ۵ انتگرال‌گیری عددی | | |
| ۱۳۱ | مقدمه | ۱.۵ |
| ۱۳۱ | انتگرال‌گیری عددی و قانون سیمپسون | ۲.۵ |
| ۱۳۴ | انتگرال‌های نامعین | ۳.۵ |
| ۱۳۶ | تابع <code>quad</code> متلب | ۴.۵ |
| ۱۳۸ | میدان الکتریکی | ۵.۵ |
| ۱۴۱ | رسم نمودار توسط <code>quiver</code> | ۶.۵ |
| ۱۴۳ | تابع <code>dblquad</code> در متلب | ۷.۵ |
| ۱۴۷ | مسائل | |
| ۱۴۷ | پروژه‌ها | |
| ۶ انتگرال‌گیری عددی معادلات دیفرانسیل معمولی | | |
| ۱۵۵ | مقدمه | ۱.۶ |
| ۱۵۵ | مسئله مقدار اولیه | ۲.۶ |
| ۱۵۵ | آلگوریتم اولر | ۳.۶ |
| ۱۵۷ | روش اولر اصلاح شده | ۴.۶ |
| ۱۶۳ | خطای عددی آلگوریتم اولر | ۵.۶ |
| ۱۶۴ | روش رانج کوتا مرتبه چهارم | ۶.۶ |
| ۱۶۶ | دستگاهِ دو معادله دیفرانسیل مرتبه اول | ۷.۶ |
| ۱۶۸ | معادله مرتبه دوم | ۸.۶ |
| ۱۷۱ | تابع معادله دیفرانسیل معمولی متلب | ۹.۶ |
| ۱۷۴ | مسائل مقدار مرزی | ۱۰.۶ |
| ۱۷۵ | حل دستگاه سه-قطري معادلات خطی | ۱۱.۶ |
| ۱۷۸ | روش تفاضلات محدود | ۱۲.۶ |
| ۱۸۰ | خازن صفحه یک بعدی | ۱۳.۶ |
| ۱۸۳ | پروژه‌ها | |

| | | |
|-----|---|-------|
| ۱۹۵ | تبدیل لاپلاس | ۷ |
| ۱۹۵ | مقدمه | ۱.۷ |
| ۱۹۵ | تبدیل لاپلاس و تبدیل معکوس | ۲.۷ |
| ۱۹۶ | تبدیل لاپلاس تابع پله واحد | ۱.۲.۷ |
| ۱۹۶ | تابع نمائی | ۲.۲.۷ |
| ۱۹۶ | خطی بودن | ۳.۲.۷ |
| ۱۹۶ | تاخیر زمانی | ۴.۲.۷ |
| ۱۹۷ | نمائی مختلط | ۵.۲.۷ |
| ۱۹۸ | توانهای t | ۶.۲.۷ |
| ۱۹۸ | تابع ضربه | ۷.۲.۷ |
| ۲۰۰ | تبدیل لاپلاس مشتق | ۳.۷ |
| ۲۰۱ | معادلات دیفرانسیل معمولی، مسائل مقدار اولیه | ۴.۷ |
| ۲۰۸ | کانولوشن | ۵.۷ |
| ۲۱۲ | کاربرد تبدیل لاپلاس در مدارها | ۶.۷ |
| ۲۱۵ | پاسخ ضربه | ۷.۷ |
| ۲۱۶ | مسائل | |
| ۲۱۷ | پروژه‌ها | |
| ۲۲۵ | تبدیل فوریه و پردازش سیگنال | ۸ |
| ۲۲۵ | مقدمه | ۱.۸ |
| ۲۲۶ | تشریح ریاضی سیگنالهای متناوب: سری فوریه | ۲.۸ |
| ۲۲۹ | سری نمائی فوریه مختلط و تبدیل فوریه | ۳.۸ |
| ۲۳۲ | خواص تبدیل فوریه | ۴.۸ |
| ۲۳۵ | فیلترها | ۵.۸ |
| ۲۳۷ | نمایش زمان گسسته سیگنالهای زمان پیوسته | ۶.۸ |
| ۲۳۹ | تبدیل فوریه سیگنالهای زمان گسسته | ۷.۸ |
| ۲۴۲ | یک فیلتر زمان گسسته ساده | ۸.۸ |
| ۲۵۰ | پروژه‌ها | |
| ۲۵۷ | برازش منحنی | ۹ |
| ۲۵۷ | مقدمه | ۱.۹ |
| ۲۵۷ | روش کمترین مربعات | ۲.۹ |
| ۲۵۷ | بهترین برازش خط مستقیم | ۱.۲.۹ |
| ۲۵۹ | بهترین برازش چندجمله‌ای درجه m | ۲.۲.۹ |
| ۲۶۰ | برازش منحنی با تابع نمائی | ۳.۹ |
| ۲۶۱ | تابع polyfit متلب | ۴.۹ |
| ۲۶۶ | خط نقطه‌گذار مکعبی | ۵.۹ |
| ۲۶۷ | تابع interp1 برای برازش خط نقطه‌گذار مکعبی | ۶.۹ |
| ۲۶۹ | برازش منحنی با سری فوریه | ۷.۹ |
| ۲۷۱ | پروژه‌ها | |
| ۲۷۵ | بهینه‌سازی | ۱۰ |
| ۲۷۵ | مقدمه | ۱.۱۰ |
| ۲۷۵ | مسائل بهینه‌سازی بدون محدودیت | ۲.۱۰ |
| ۲۷۶ | روش تندترین نزول | ۳.۱۰ |

| | | |
|------------|--|-----------|
| ۲۸۰ | تابع fminunc در متلب | ۴.۱۰ |
| ۲۸۱ | بهینه‌سازی با محدودیت | ۵.۱۰ |
| ۲۸۳ | ضرب کننده‌های لاگرانژ | ۶.۱۰ |
| ۲۸۵ | تابع fmincon متلب | ۷.۱۰ |
| ۲۹۴ | مسائل | |
| ۲۹۴ | پروژه‌ها | |
| ۳۰۱ | سیمیولینک | ۱۱ |
| ۳۰۱ | مقدمه | ۱.۱۱ |
| ۳۰۱ | تولید یک مدل در سیمیولینک | ۲.۱۱ |
| ۳۰۳ | نمونه‌ای از بلوک‌های ساختمانی در ساخت یک مدل | ۳.۱۱ |
| ۳۰۵ | نکاتی چند در ساختن و اجرای مدل‌ها | ۴.۱۱ |
| ۳۰۷ | ایجاد یک زیرسامانه | ۵.۱۱ |
| ۳۰۸ | استفاده از بلوک‌های Fcn و Mux | ۶.۱۱ |
| ۳۰۸ | استفاده از انتقال بلوک Fcn | ۷.۱۱ |
| ۳۰۹ | استفاده از بلوک‌های سویچ و رله | ۸.۱۱ |
| ۳۱۱ | بلوک‌های توابع مثلثاتی | ۹.۱۱ |
| ۳۱۳ | مسائل | |
| ۳۱۵ | پروژه‌ها | |

فصل ۱

روش‌های عددی برای مهندسین برق

۱.۱ مقدمه

تمام نظام علمی و مهندسی از روش‌های عددی برای تحلیل‌های مسائل پیچیده استفاده می‌کنند. البته، مخصوصاً مهندسین برق، نظر خود را بر حل محاسباتی ناشی از طبیعت ریاضی میدانی و رابطه نزدیک آن با علوم کامپیوتر معطوف می‌کنند. ببین است که مهندسینی که کامپیوتراهای پر سرعت را طراحی می‌کنند، از کامپیوتر برای طراحی خود استفاده می‌نمایند. (این فرآیند خود راه انداز^۱ معروف است) در واقع، تمام عرصه طراحی بكمک کامپیوتر^۲ CAD مدیون تولید و پیشرفت برنامه نرم‌افزاری است که توانائی پیاده‌سازی طراحی‌های بسیار پیچیده را میسر و فراهم می‌کند.

این کتاب روش‌ها و فناوری‌های مختلف را برای حل مسائل گوناگون که در مهندسی برق کاربرد دارند، از قبیل طراحی مدار، نظریه میدان‌های الکترومغناطیسی، و پردازش سیگنال‌ها توصیف می‌کند. در برنامه تحصیلی مهندسی روش‌های مختلف را برای حل اینگونه مسائل با استفاده از فناوری‌هایی از قبیل جبر خطی، معادله دیفرانسیل، تبدیل‌های مختلف، ریاضیات برداری و غیره آموزش میدهدن. البته، در هر حالت، کاوش برای حل فرم بسته منجر به معضلات زیادی شده، که می‌تواند ما را از درک مفاهیم فیزیکی دور کند. در حل همان مسئله بروش عددی، اکثرًا ما را به روابط فیزیکی بنیادین، مانند رابطه دیفرانسیلی (جزئی) بین جریان حاضر و ولتاژ یا میدان الکتریکی بار نقطه‌ای، هدایت می‌کند. اغلب، وقتی مسئله‌ای بطور سمبلیک حل شود ممکن است رام نشدنی و مشکل بنظر برسد، ولی وقتی بطور عددی حل شود، می‌تواند بدیهی و آسان شود. و بعضی وقت‌ها حل یک مسئله ساده عددی

^۱bootstrapping

^۲computer-aided design (CAD)

می‌تواند گریزان و فراری شود زیرا ما عادت داریم در حوزه ریاضیات عالی فکر کنیم (حالت کلاسیک این است که نتوانیم جنگل را ببینیم، اما نظر ما به درختان باشد).

۲.۱ اهداف مهندسی

برخی از اهداف بنیادین در مهندسی بقرار زیر است:

- محصول جدیدی طراحی کردن یا نوع موجود را ارتقاء دادن.

- کارائی محصولی را افزایش دادن.

- هزینه، توان مصرفی، و هزینه مهندسی غیر قابل برگشت^۳ را کمینه کردن.

- افزایش بازده و بازگشت سرمایه^۴

- کمینه کردن زمان در بازار

غالباً مهندسین قوانین فیزیکی و ریاضیات را برای نیل بدین اهداف بکار می‌برند.

خیلی از تحولات مهندسین برق شامل مراحل ساخت پرهزینه بوده که هم ظریف و هم زمان بر است. برای مثال، ساخت مدارات مجتمع شامل هزاران مرحله ساخت شامل، تهیه ویفر^۵، تولید ماسک، لیتوگرافی نوری، نفوذ و کاشت، برش و قطعه قطعه کردن، آزمایش، بسته‌بندی، وغیره است. این مراحل می‌تواند هفت‌ها یا ماه‌ها با هزینه‌های زیاد اطاقهای پاک^۶ انجام شود. هر اشتباه طراحی لازمه تکرار مراحل است که کار ما بعنوان طراح، مدل سازی و شبیه‌سازی طرح تا حد امکان در پیشرفت ساخت و حذف خدشه و عیب و کمینه کردن تکرار ساخت لازم برای محصول نهایی است. برای مثال، برای طراحی مدارات مجتمع، باید از کامپیوتر بمنظور مراحل زیر استفاده شود.

- الف- مرحله طراحی: حل مدل ریاضی پدیده‌های فیزیکی (یعنی؛ پیش‌بینی رفتار اتصالات PN)

- ب- مرحله آزمایش: ذخیره سازی و تحلیل داده‌های آزمایش (یعنی؛ تهیه آزمایشگاه اندازه‌گیری رفتار واقعی اتصالات PN نسبت به پیش‌بینی)

- ج- مرحله ساخت: کنترول عملکرد ماشین‌ها در ساخت و آزمایش ویفر سیلیکون و قطعات کوچک.

^۳ nonreturnable engineering (NRE)

^۴ return on investment (ROI)

^۵ wafer

^۶ clean rooms

۳.۱ برنامه‌نویسی حل عددی

پدیده‌های فیزیکی همیشه با مجموعه‌ای از معادلات حاکم توصیف شده، و روش‌های عددی را می‌توان به منظور حل مجموعه‌ای از معادلات حاکم، حتی در غیاب جواب فرم بسته، استفاده کرد. روش‌های عددی منفک از کامپیوتر نبوده، و کامپیوتر محاسبات ریاضی را بصورت اعداد گسسته در مراحل تعریف شده‌ای اجرا می‌کند. مراتب مراحل در برنامه تعریف می‌شود. جواب مناسب وقتی بدست می‌آید که

- الف- مدل ریاضی دقیقاً نمایشگر پدیده‌های فیزیکی باشد؛ یعنی، مدل معادلات حاکم درستی داشته باشد.

- ب- روش عددی دقیق باشد.

- ج- روش عددی دقیقاً برنامه‌نویسی شود.

این کتاب اساساً به اقلام (ب) و (ج) مربوط می‌شود.

مزایا استفاده از کامپیوتر این است که آن می‌تواند تعداد معنایه‌های محاسبه را در کسری از ثانیه انجام دهد؛ در موقع نوشتمن این مطالب، سرعت کامپیوتر در ترافلاب (ملياردها عملیات نقطه شناور در ثانیه) اندازه‌گیری شده است. البته، برای بکارگیری این توان، به نوشتمن مجموعه‌ای دستورالعمل، که به آن برنامه‌نویسی می‌گویند نیازمندیم. برای مسائل این کتاب، یک کامپیوتر دیجیتالی که فقط قادر باشد اعمال محاسباتی، منطقی، و عملیات گرافیکی را انجام دهد مطلوب ما و کافی است. بنابراین، شیوه محاسبه برای حل معادلات دیفرانسیلی، که شامل محاسبه انتگرال، تعیین ریشه یک معادله، حل یک دستگاه معادلات خطی، وغیره باید مدون و نگاشته شود. نحوه محاسبه، عموماً شامل مجموعه‌ای از معادلات جبری است. حل کامپیوتری برای اینگونه مسائل شامل تدوین یک برنامه کامپیوتری است که قدم به قدم روند تعیین جواب مسئله مطلوب ما تعریف شده است. این روش حل بنام **آلگوریتم^۷** خوانده می‌شود. برای هر مسئله خاص، میباشد آلگوریتم خودمان را بنویسیم یا همان‌طوری که خواهیم دید، می‌توانیم از آلگوریتم‌هایی مانند آنچه در داخل نرم‌افزار متلب MATLAB، از قبیل آلگوریتم مشهور رانج-کوتا^۸ برای حل معادلات دیفرانسیل معمولی یا روش سیمپسون^۹ برای محاسبه انتگرال‌ها، که نوشته شده است استفاده کنیم.

۴.۱ چرا متلب؟

متلب MATLAB اولیه توسط دکتر کلومولر در دانشگاه نیومکزیکو در سال 1970 نوشته و در سال 1980 توسط شرکت متورکس MathWorks تجاری شد. این یک بسته ریاضی برای کاربردهای عمومی

^۷algorithm

^۸Runge-Kutta

^۹Simpson's rule

است که بما اجازه می‌دهد معادلات پیچیده را بنحو موثری حل کرده و در خروجی جدولی از ارقام و نمودارهای گرافیکی تحويل می‌دهد. با وجودی که تعداد زیادی بسته‌نرم‌افزاری برای مهندسین برق وجود دارد، اما برخی از آنها برای کاربردهای خاصی (از قبیل اسپایس برای مدل‌سازی مدارات الکترونیکی) متمرکز و تدوین شده‌اند. هم‌چنین، متلب روی نرم‌افزار طراحی بكمک کامپیوتر CAD مانند رسم مدار، مدار چاپی، طراحی فیزیکی، متمرکز نشده اما این نرم‌افزار اغلب با بسته‌های نرم‌افزاری عددی مجتمع و همراه شده است.

ابتدا مطلب بصورت برنامه خط-دستوری که در سیستم عامل MS-DOS و UNIX اجرا می‌شد. با پیشرفت کامپیوتراها، مطلب نیز کسترش یافته و ویرایش‌های جدید در محیط ویندوز اجرا شد. در حال حاضر، مطلب R2011b روی ویندوز میکروسافت، آپل، و لینوکس اجرا می‌شوند. در این کتاب، فرض ما بر این است که شما مطلب را روی کامپیوتر محلی خودتان تحت محیط ویندوز میکروسافت اجرا می‌کنید. در هر حالت، اختلاف آنها بیشتر در آرایش و نمایش برنامه است و دستورات مطلب فرقی نمی‌کنند. تمام تنوع‌های مطلب (در هر سیستم عاملی) مجموعه دستورات یکسانی را استفاده می‌کند و فرمان ویندوز روی تمام سیستم عامل‌ها رفتاری مشابه‌ای را باید داشته باشند.

نرم‌افزار مطلب با "جعبه‌ابزار"‌هایی^{۱۰} با هزینه اضافی برای مصرف کننده همراه هست. تنوع جعبه‌ابزارها در زمینه‌هایی از قبیل سیستم‌های کنترول، پردازش تصویر، طراحی مدارهای رادیوئی (RF)، پردازش سیگنال، و غیره موجود است. البته، در این کتاب، بیشتر تمرکز ما بر مفاهیم عددی بنیادی و نظر خود را به توابع پایه مطلب بدون نیاز به خرید جعبه‌ابزارهای اضافی معطوف خواهیم کرد.

۵.۱ زبان برنامه نویسی مطلب

روشهای متعددی برای برنامه‌نویسی کامپیوتر هست، اما کارها در حال حاضر در اطراف مطالب زیر است:

- الف- مطالعه مسئله‌ای که باید برنامه‌نویسی شود.
- ب- فهرست نمودن معادلات جبری بر مبنای پدیده‌های فیزیکی معلوم و هندسه مسئله که در برنامه باید بکار رود.
- ج- برای روند برنامه نویسی و آلگوریتم طرح کلی ترسیم کرده و شاید با ترسیم یک نمودار جریان^{۱۱} یا نوشتן کد پیشرفت‌های هسته اصلی برنامه بوجود آورید.
- د- با یک محاسبه ساده دستی آلگوریتم برنامه را اثبات کنید.
- ه- با استفاده از فهرستی از معادلات جبری و روئوس مطالب برنامه را بنویسید.

^{۱۰} toolboxes

^{۱۱} flowchart

• و- با جرا آن توسط پارامترهای با جواب معلوم، برنامه را آزمایش کنید.

• ز- این مراحل را تکرار کرده تا نمودار جریان و الگوریتم را اشکال‌زدایی^{۱۲} و پالایش نمایید.

• ح- اگر لازم باشد، برنامه را تجدید نظر کرده تا اجرای سریعتری حاصل شود.

برنامه‌نویس‌های با تجربه ممکن است برخی از این مراحل را حذف نمایند (و شاید هم آنها را در ذهن خود انجام دهند)، اما فرایнд کلی برای هر پروژه مهندسی شبیه همین است: طراحی، ساخت یک نمونه اولیه^{۱۳}، آزمایش، و تکرار فرآیند تا نتیجه مطلوب بدست آید.

متلب را ممکن است یک محیط برنامه‌نویسی یا زبان متنی در نظر گرفت، اما مانند هر زبان برنامه‌نویسی، دارای مولفه‌های اصلی زیر است:

• الف- انواع داده‌ها؛ یعنی، شکلی از اعداد و حروف در برنامه (مثل، اعداد صحیح، اعداد اعشاری با دقت مضاعف، رشته‌ها^{۱۴}، بردارها، ماتریس‌ها)

• ب- عملگرها (مثل، فرامین جمع، ضرب، کسینوس، لگاریتم)

• ج- روند کنترول توجیهی برای تصمیم‌گیری و انجام کاربردهای تکراری (مثل، if، while، .switch).

• د- فرامین ورودی-خروجی ("I/O") برای استفاده از ورودی کاربر یا فایل و برای تولید خروجی به فایل یا صفحه نمایشگر (مانند، fprintf، fscanf، plot، stem، surf).

متلب خیلی از اینها را از سایر زبان‌ها قرض گرفته است. برای مثال، switch، while، fprintf، و switch را از زبان برنامه‌نویسی C (یا نسل‌های بعدی آن C++، Java، Perl) . البته، اختلافات بنیادی نیز وجود دارد. برای مثال، در متلب توابع functions (که در سایر زبانها بدان "زیر برنامه" یا سابروتین subroutines نامیده می‌شود) در فایل جداگانه ذخیره می‌شود. اولین جمله در یک بردار vector (که در اکثر دیگر زبانها آن را آرایه array می‌نامند) با شماره یک (نه صفر) شماره‌گذاری می‌شود. البته، بزرگترین اختلاف این است که متلب متغیرها برداری هستند، بنابراین این توانائی را دارد که مقادیر زیادی از داده را با این سبک و سیاق دستکاری و عملیات انجام دهد و فقط با نوشتن چند خط کد، بدنبال جواب مسائل پیچیده باشد. بعلاوه، چون متلب بطور عادی همراه هم و اجرا می‌شود، در نمایش توابع پیچیده و نمودارها غنی است.

^{۱۲}debug

^{۱۳}prototype

^{۱۴}strings

۶.۱ قراردادهای این کتاب

در این کتاب ما از قراردادهای زیر استفاده می‌کنیم:

- تمام ورودها به متلب و خروجی‌ها از متلب بصورت فونت ماشین تحریری است.
- در حالتی که شما مستقیماً در کامپیوتر تایپ می‌کنید، با حروف درشت^{۱۵} نمایش داده می‌شود.

این را با مثال زیر که در متلب برای پیدا کردن مقدار $x = \sin \frac{\pi}{4}$ بکار میبریم، روش خواهیم کرد:

```
>> x = sin(pi/4)
x =
0.7071
```

در این حالت، >> نمایشگر آمادگی متلب، $x = \sin(\text{pi}/4)$ نمایشگر فرمان متلب و ۰.۷۰۷۱ نمایشگر جواب متلب است.

۷.۱ برنامه‌های نمونه

برنامه‌های نمونه این کتاب را می‌توان از درگاه ناشر

<http://www.crcpress.com/product/isbn/9781439854297>

دسترسی کنید. دانشجویان می‌توانند برنامه‌های نمونه را روی کامپیوتر خود اجرا و نتایج را مشاهده نمایند. ممکن است برای دانشجویان برخی برنامه‌ها را خود تایپ کرده (که همراه این کار اشتباهاتی ممکن است رخ دهد) از اینرو فرستی پیش می‌آید که بهبینید چطور متلب به غلط‌های برنامه شما پاسخ میدهد و در نتیجه شما یاد می‌کیرید که چگونه برنامه خود را اصلاح نمائید.

^{۱۵}bold

فصل ۲

مبانی متلب (MATLAB)

۱.۲ مقدمه

متلب (MATLAB) یک برنامه نرم افزاری برای محاسبات ریاضی، تحلیل داده، و ترسیم است. یکی از نتایجی که متلب برای مهندسین نسبت به سایر زبان های برنامه نویسی از قبیل C++ یا C دارد این است که متلب شامل توابعی است که می تواند دستگاه معادلات خطی بزرگ، معادلات دیفرانسیل معمولی، ریشه های معادلات متعالی^۱، انگرال ها، مسائل آماری، مسائل بهینه سازی، مسائل سیستم های کنترول، و خیلی دیگر از مسائل که در مهندسی با آن مواجه می شویم را از نظر عددی حل کند. ضمناً متلب جعبه ابزارهای نیز ارائه نموده است (آنها را میباید جداگانه خریداری کرد) که آنها برای حل مسائل خاص طراحی شده اند.

در این فصل، اقلام زیر را پوشش میدهیم:

- میز کار متلب.
- تولید متن (آنرا برنامه نیز میگویند) در متلب.
- مبانی متلب و فرامین اصلی، شامل `clc`, `clear` عملگر دونقطه^۲، عملگرهای محاسباتی، توابع مثلثاتی، تابع نمائی و لگاریتمی، و سایر توابع مفید دیگر از قبیل بیشینه `max` کمینه `min` و طول `.length`.
- ورودی / خروجی در متلب، شامل بیانیه های ورودی `input` و چاپ `fprintf`

¹transcendental equations

²colon operator

- روند برنامه متلب، شامل بیانیه‌های `for` برای حلقه‌ها، حلقه‌های شرطی `if`، `اگرها`، `و` `elseif`، `و` `بیانیه گروه کلید switch`.
- فایل‌های تابع متلب و توابع مستعار.
- کرافیک متلب، شامل فرامین `plot` و `subplot`.
- کار با ماتریس‌ها.
- کار با توابع و بردارها.
- کار با هر نوع حروف نوشتنی و چاپی^۳ و رشته‌ها^۴.
- درون‌یابی^۵ و تابع درون‌یاب `interp1` در متلب.
- تابع `textscan` در متلب.
- صادر کردن داده متلب به نرم‌افزار دیگر، از قبیل اکسل مایکروسافت.
- برنامه اشکال‌زدایی^۶.

خیلی از این گفته‌ها در سراسر این فصل آورده شده است تا اینگونه عنایوین مختلف را روشن نماید.

۲.۲ پنجره‌های متلب

تحت پنجره‌های مایکروسافت، متلب را می‌توان از طریق منو Start یا فشردن (کلیک) آیکن متلب روی میز کار آنرا شروع کرد. در این موقع پنجره جدیدی که شامل محیط متلب است باز می‌شود. به شکل (۲.۱) مراجعه کنید تا محیط مورد نظر را مشاهده کنید. پنجره‌های اصلی، پنجره فرمان، گذشته فرامین^۷، پوشه فعلی^۸، و فضای کار^۹ هستند. می‌توانید پنجره‌های متلب را در موقع باز شدن روی منوی میز کار، مطابق میل خود تنظیم کنید. شکل (۲.۱) پنجره فرمان (در وسط)، پوشه فعلی (در سمت چپ)، فضای کار (در بالای سمت راست)، گذشته فرامین (سمت راست پائین)، و جعبه پوشه فعلی (در آیکن نوار ابزار، دومی از بالا، درست بالای پنجره فرمان) را نشان میدهد. این پنجره‌ها و جعبه پوشه فعلی چنین خلاصه می‌شود:

^۷characters

^۸strings

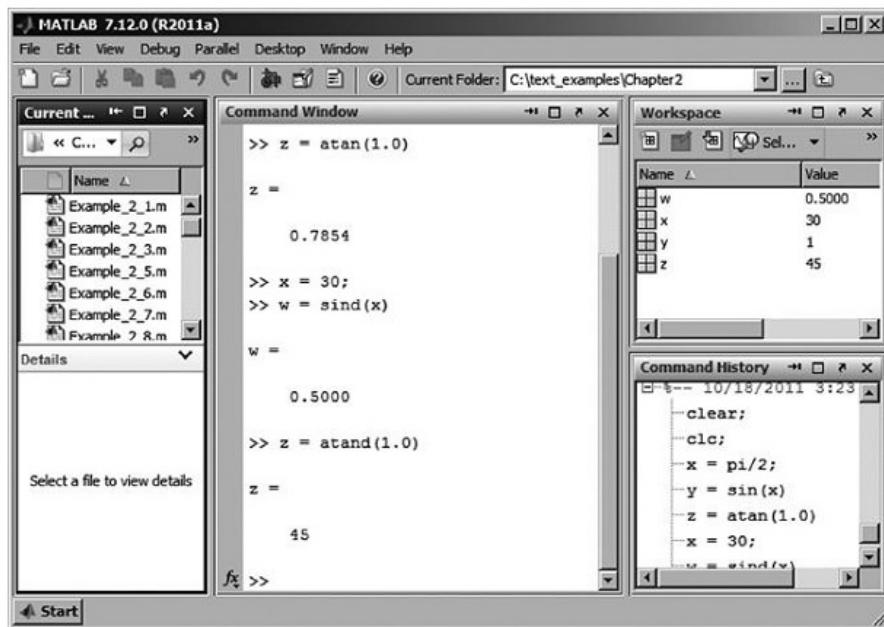
^۹Interpolation

^{۱۰}Debugging

^{۱۱}Command History

^{۱۲}Current Folder

^{۱۳}Workspace



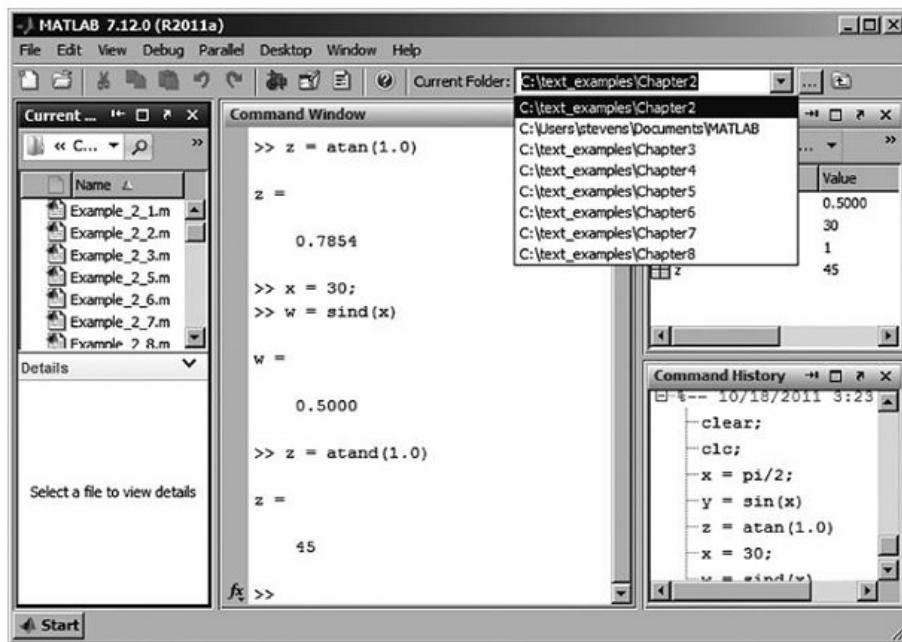
شکل ۱.۲: پنجره متلب روی میزکار

■ پنجره فرمان: در پنجره فرمان می‌توانید فرمان‌ها و داده‌ها را وارد کرده، محاسبات ریاضی را انجام داده و نتایج را چاپ نمائید. می‌توانید در پنجره فرمان، یک برنامه نوشته و برنامه را اجرا و محاسبه کنید. البته، نوشتن یک برنامه مستقیماً در پنجره فرمان مایوس کننده است زیرا نمی‌توانید آنرا نکهدارید و ذخیره save نمایید. و اگر یک خطای داشته باشید، میبایست تمام برنامه را دوباره تایپ کنید. با استفاده بردار (\uparrow) روی صفحه کلید، می‌توانید فرمان قبلی را بیاورید (ویرایش کنید) و دوباره اجرا نمایید.

■ پنجره گذشته فرامین: این پنجره فرامین گذشته‌ای که شما در پنجره فرمان اجرا کردید را فهرست می‌کند.

■ جعبه پوشه فعلی:^{۱۰} این جعبه پوشه فعلی فعال (در متلب‌های قدیمی بدان دایرکتوری فعلی می‌نامیدند) را فهرست می‌کند. برای اجرای متن (برنامه)، متن مورد نظر باید در پوشه این جعبه باشد. با کلیک روی بردار روبه پائین درون این جعبه، یک منو ظاهر شده که شامل اسامی پوشه‌هایی است که شما قبلاً بکار برده‌اید. این بشما اجازه را میدهد تا پوشه‌ای که در آنجا قرار دارد انتخاب کنید (به شکل ۲.۲ مراجعه کنید).

^{۱۰} Current Folder Box



شکل ۲.۲: پائین آوردن منو در پنجره پوشه فعلی

■ **پنجره پوشه فعلی^{۱۱}** (در سمت چپ): این پنجره تمام فایل‌های درون پوشه فعلی را فهرست می‌کند. با دو مرتبه کلیک روی فایل در این پنجره، فایل در متلب باز می‌شود.

■ **پنجره متن** (در متلب قدیم، این بنام پنجره ویرایشگر^{۱۲} نیز خوانده می‌شد): برای باز کردن این پنجره، منو فایل File در بالای میزکار متلب استفاده کرده، و سپس متن Script (در متلب‌های قدیم روی New M-File) کلیک کنید. (به شکل ۴.۲ مراجعه کنید). پنجره متن را می‌توانید برای تولید، ویرایش، و اجرای متون متلب استفاده کنید. سپس متن بصورت ام‌فایل M-File ذخیره می‌شود. این فایلها پسوند m. را مثل circuit.m همراه دارند. برای اجرا برنامه، می‌توانید از آیکن Save and Run در پنجره متن استفاده کرده یا به پنجره فرمان رفته و نام برنامه را (بدون پسوند m.) تایپ کنید.

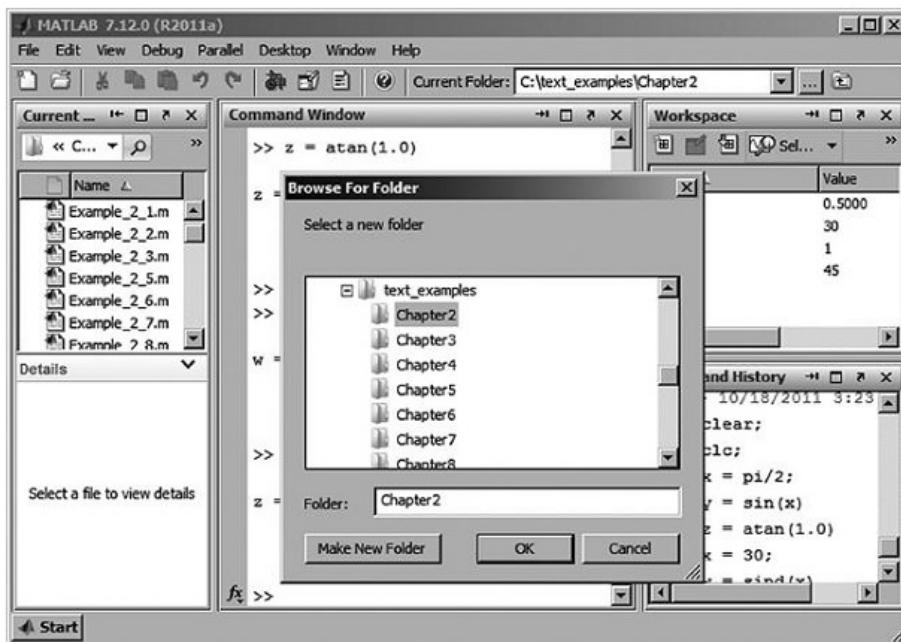
۳.۲ نوشتن برنامه در متلب

در زیر خلاصه‌ای از مراحل نوشتن اولین برنامه متلب شما را می‌آوریم:

۱. متلب میزکار را از طریق منو شروع پنجره یا با دوبار کلیک روی آیکن متلب در روی میزکار خود

^{۱۱}Current Folder Window

^{۱۲}Editor



شکل ۳.۲: پائین آوردن منو برای انتخاب پوشاهای که برنامه مورد نظر آنجا است.

شروع کنید.

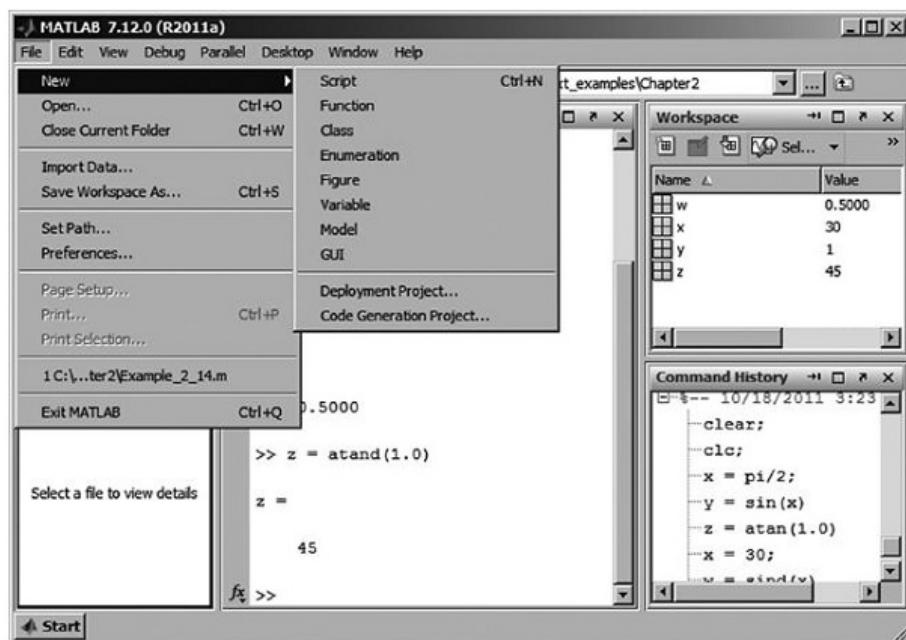
۲. روی *File-New-Script* کلیک کنید. این یک پنجره متن (نوشتاری) برای شما باز می‌کند.

۳. برنامه خود را در این پنجره متن تایپ کنید.

۴. برنامه را با کلیک کردن روی آیکن *Save* در میان آیکن نوار ابزار ذخیره نمایید. البته می‌توانید با کلیک روی *File* در منو که به پائین کشیده می‌شود، *Save* را انتخاب کرده و آنرا ذخیره نمایید. در جعبه‌ای که ظاهر می‌شود، پوشاهای که می‌خواهید در آن پوشه فایل شما قرار گیرد، نام فایل خود را تایپ کنید. بهتر است از پوشاهای استفاده کنید که فقط فایل‌های متلب شما است.

۵. قبل از آنکه بتوانید برنامه خود اجرا کنید، لازم است به جعبه پوشه فعلی خود در بالای میزکار متلب رفته، روی پیکان رویه پائین کلیک کرده و در منوئی که رو به پائین آورده شده، پوشاهای که برنامه جدید را دارد انتخاب کنید.

۶. می‌توانید برنامه خود را در پنجره متن با کلیک روی *Save and Run* با پیکان سبز رنگ در آیکن نوار ابزار toolbar (به شکل ۵.۲ مراجعه کنید) اجرا کنید. ضمناً راه دیگر این که می‌تواند در پنجره فرمان با تایپ نام برنامه خود (بدون پسوند.m) بعد از علامت متلب (>>) اجرا نمایید. برای مثال، اگر برنامه شما بنام circuit.m ذخیره شده باشد، بعداز علامت (>>) کلمه



شکل ۴.۲: باز کردن پنجره متن.

را تایپ کنید. مانند آنچه در زیر نشان داده شده است:

```
>> circuit
```

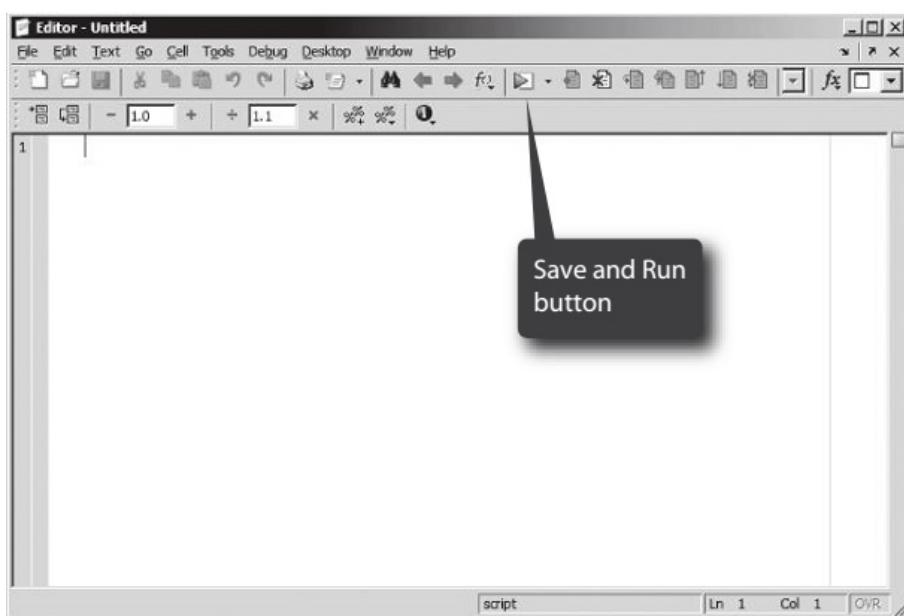
اگر به کمک اضافی نیاز داشتید، می‌توانید روی Help نوار منو در پنجره متلب کلیک کرده و سپس جمله Produce Help را انتخاب کنید. این در پنجره متلب مطابق شکل (۶.۲) خواهد بود. با کلیک روی جعبه '+' کوچک نزدیک ستون سمت چپ، مانند شکل (۷.۲)، اطلاعات کمکی اضافی در پنجره دست راست خواهد داشت. ضمناً، می‌توانید با تایپ کردن در پنجره جستجو اطلاعاتی در این زمینه بدست آورید.

۴.۲ مبانی متلب

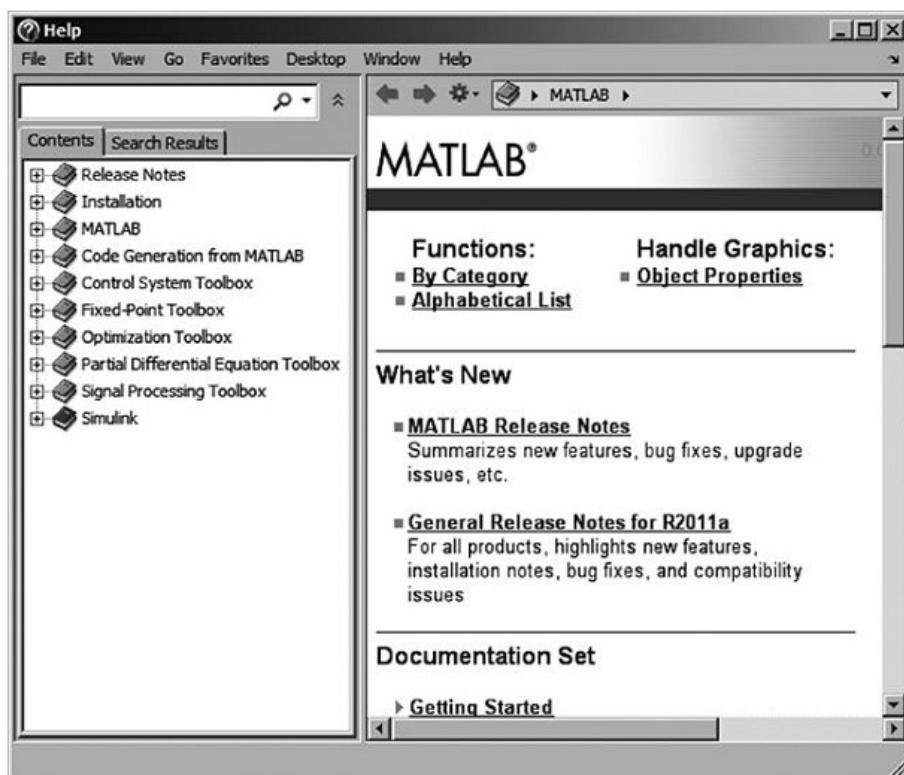
نام متغیرها ■

- باید با حروف الفبائی شروع شود.
- می‌تواند شامل حروف، اعداد، و مشخصه زیرنشان^{۱۳} باشد.

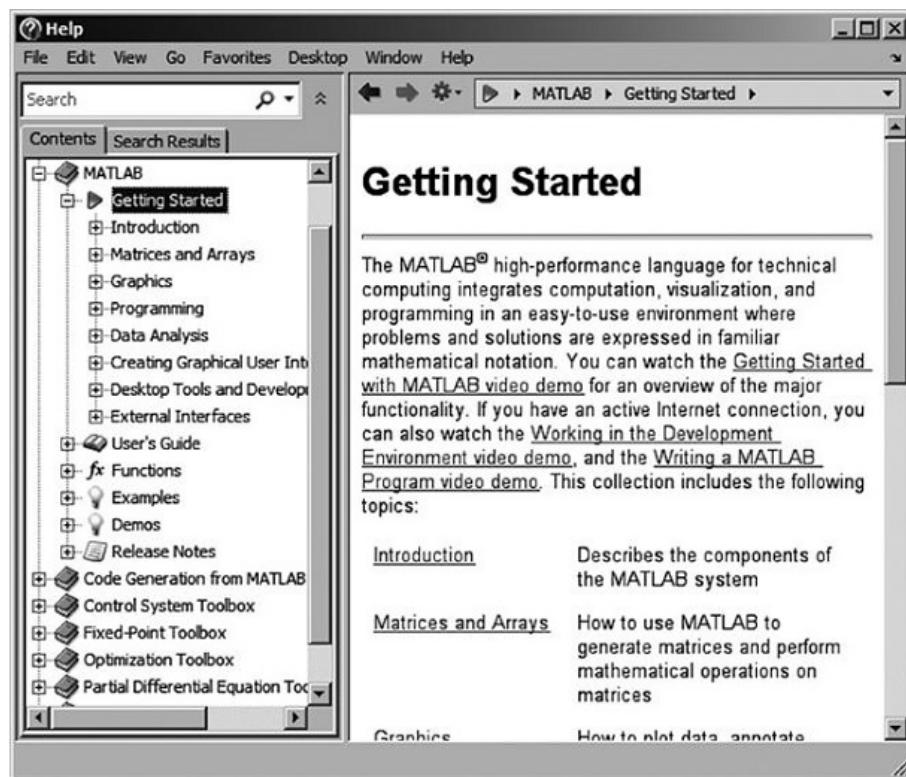
^{۱۳}under score character



شکل ۲.۵: ذخیره و اجرا در پنجره متن.



شکل ۲.۶: تولید پنجره امداد



شکل ۷.۲: شروع در پنجره امداد

- می‌تواند هر طولی داشته باشد، اما باید در مشخصه نوزده حرف اول باشد.

توجه: از بکار بردن نام متغیرهایی که بصورت نام فایل، تابع متلب، یا یک تابع نوشته شده خودداری کنید.

■ نام فرامین متلب و نام متغیرها نسبت به درشت یا ریز بون حروف حساس هستند. برای فرامین از حروف کوچک استفاده کنید.

■ واک و نقطه^{۱۴} (;) معمولاً بعداز تعریف متغیرها و بیان برنامه وقتی میخواهیم فرمان روی صفحه نمایش داده نشود، استفاده می‌کنیم. در غیاب واک و نقطه (;)، تعریف متغیرها روس صفحه ظاهر می‌شود. برای مثال، اگر فرمان زیر در پنجره فرمان وارد شود:

```
>> A = [3 4 7 6]
```

^{۱۴}Semicolon

در پنجره، خواهید دید

```
A =
3 4 7 6
>>
```

برعکس، اگر واک و نقطه اضافه شود، فرمان شما اجرا خواهد شد اما روی صفحه نمایشگر چیزی چاپ نخواهد شد. و فقط علامت آمادگی اجرا > برای فرمان بعدی شما ظاهر خواهد شد:

```
>> A = [3 4 7 6];
>>
```

- علامت درصد (%) برای خط توضیح استفاده می‌شود.
- برای نمودارها و گرافیک پنجره گرافیک جداگانه‌ای باز می‌شود.
- چندین فرمان برای پاک کردن پنجره‌ها، پاک کردن فضای کار، و توقف اجرای برنامه وجود دارد.

| | |
|---------|--|
| c1c | پاک کردن پنجره فرمان |
| clf | پاک کردن پنجره گرافیک |
| clear | پاک کردن تمام متغیرها و داده‌ها از فضای کار |
| Ctl – C | برای توقف برنامه که یک حلقه بینهایت را اجرا می‌کند |

- فرمان‌های quit یا exit برنامه متلب را پایان میدهد.
- فرمان save متغیرها یا داده‌ها را در فضای کار در دایرکتوری فعلی ذخیره می‌کند. نام فایلی که حاوی داده باشد دارای پسوند .mat است.
- توابعی که کاربر تعريف می‌کند (آنرا توابع خود نوشته می‌نامیم) بصورت ام-فایل ذخیره می‌شود.
- توابع و متون بصورت فایل ASCII ذخیره می‌شوند. بنابراین، میتوان آنها را در پنجره متن، نوت‌پد^{۱۵}، یا هر پردازشکر نوشتاری ساخت و بصورت فایل متند^{۱۶} ذخیره کرد.
- ساختار داده اصلی در متلب ماتریس است.
- یک ماتریس با دو براکت محدود می‌شود و ممکن است دارای هر تعداد دلخواهی سطر و ستون داشته باشد؛ برای مثال، ماتریس $A = \begin{matrix} 1 & 3 \\ 6 & 5 \end{matrix}$ در متلب را میتوان بصورت زیر وارد کرد.

```
>> A = [1 3 <enter>
       6 5] ; <enter>
```

^{۱۵}Notepad

^{۱۶}text file

یا

```
>> A = [1 3 ; 6 5]; <enter>
```

که در آن واوک و نقطه در درون براکت نشانده شروع سطر جدید درون ماتریس است.

■ ماتریس یک سطر و یک ستون برابر یک عدد اسکالر است، مثلًاً

```
>> A = [3.5];
```

نوع دیگر، متلب نیز $A = 3.5$ (بدون براکت) را عنوان عدد اسکالر قبول می‌کند.

■ ماتریسی که شامل یک سطر و چندین ستون یا یک ستون و چندین سطر باشد عنوان بردار در نظر گرفته می‌شود؛ مثلًاً

```
>> A = [2 3 6 5] (row vector)
```

```
>> A = [2  
3  
6  
5] (column vector)
```

یک ماتریس را می‌توان با ماتریس دومی عنوان یکی از عناصر آن تعریف کرد؛ مانند:

```
>> B = [1.5 3.1];  
>> C = [4.0 B]; (thus C = [4.0 1.5 3.1])
```

■ یک عنصر از ماتریس C را می‌تواند با نوشتن بصورت زیر انتخاب کرد

```
>> a = C(2); (thus a = 1.5)
```

اگر می‌خواهید آخرین عنصر در یک بردار را انتخاب کنید، می‌توانید بنویسید

```
>> a = c(end); (thus a = 3.1)
```

■ عملگر دو نقطه (:) را می‌توان برای تولید یک ماتریس جدید از یک ماتریس موجود استفاده کرد،

مثلًاً:

$$A = \begin{matrix} 5 & 7 & 10 \\ 2 & 5 & 2 \\ 1 & 3 & 1 \end{matrix}$$

$$x = A(:, 1) \quad \text{می‌دهد} \quad x = \begin{matrix} 5 \\ 2 \\ 1 \end{matrix} \quad \text{در این صورت}$$

دونقطه در رابطه $(A(:,1))$ بمعنی تمام سطرها در ماتریس A، و ۱ بمعنی ستون اول است.

$$\begin{array}{ccccc} & & & 7 & 10 \\ \mathbf{x} = \mathbf{A}(:,2:3) & \text{میدهد} & x = & 5 & 2 \\ & & & 3 & 1 \end{array}$$

دونقطه اول در رابطه $(A(2:3,:))$ بمعنی تمام سطرها در A، و ۳ : ۲ بمعنی ستون های ۲ و ۳ ضمناً می توان نوشت

$$y = A(1,:), \text{ which gives } y = [5 \ 7 \ 10]$$

عدد ۱ بمعنی اولین سطر، و علامت دونقطه بمعنی تمام ستون ها است.

■ یک دونقطه را می توان برای تولید یک سری اعداد بکار برد. شکل نوشت آن چنین است:

$$n = \text{اندازه گامها} : \text{مقدار شروع}$$

اگر اندازه گامها حذف شود، بطور خودکار اندازه گامها ۱ در نظر گرفته می شود. برای مثال:

$$n = 1 : 8 \quad n = [1 \ 2 \ 4 \ 5 \ 6 \ 7 \ 8] \quad \text{میدهد}$$

با تعیین ۲ برای گامها، خواهیم داشت

$$n = 1 : 2 : 8 \quad n = [1 \ 3 \ 5 \ 7] \quad \text{میدهد}$$

این نوع روابط اغلب برای حلقه for که بعداً در این فصل تعریف می شود استفاده خواهد شد.

■ عملگرهای محاسبه

| | |
|---|-------|
| + | جمع |
| - | تفريق |
| * | ضرب |
| / | تقسيم |
| ^ | توان |

■ برای نمایش مقدار یک متغیر، فقط نام متغیر را بدون واک و نقطه تایپ کرده تا متغیر رو صفحه نمایشگر ظاهر شود.

تمرین (سعی کنید مطالب زیر را در پنجره فرمان چاپ کنید):

```
clc;
x = 5;
y = 10;
z = x + y
w = x - y
z = y/x
z = x*y
z = x^2
```

توجه کنید که بیان ریاضی $z=x+y$ و y در دو سطر بالاتر مقداردهی شده است. بطور کلی، تمام مقادیر در سمت راست بیانهای ریاضی باید مقدار داده شدهای قبل از استفاده داشته باشند.

■ مقادیر خاص:

| | |
|------------|-------------|
| <i>pi</i> | π |
| <i>i</i> | j |
| <i>inf</i> | $\sqrt{-1}$ |
| | ∞ |

آخرین نتیجه بدون مقداردهی که در پنجره فرمان تایپ شده است.
تمرین (سعی کنید مطالب زیر را در پنجره فرمان تایپ کنید):

$$x = pi;$$

$$z = x/0 \quad \text{بینهایت میشود}$$

■ توابع مثلثاتی:

| | |
|-------|-------------------------------|
| sin | sine |
| sinh | hyperbolic sine |
| asin | inverse sine |
| asinh | inverse hyperbolic sine |
| cos | cosine |
| cosh | hyperbolic cosine |
| acos | inverse cosine |
| acosh | inverse hyperbolic cosine |
| tan | tangent |
| tanh | hyperbolic tangent |
| atan | inverse tangent |
| atan2 | four-quadrant inverse tangent |
| atanh | inverse hyperbolic tangent |
| sec | secant |
| sech | hyperbolic secant |
| asec | inverse secant |
| asech | inverse hyperbolic secant |
| csc | cosecant |
| csch | hyperbolic cosecant |
| acsc | inverse cosecant |
| acsch | inverse hyperbolic cosecant |
| cot | cotangent |
| coth | hyperbolic cotangent |
| acot | inverse cotangent |
| acoth | inverse hyperbolic cotangent |

آرگومان این توابع مثلثاتی بر حسب رادیان است. البته، اگر حرف "d" بعدا از نام تابع قرار دهیم، از قبیل $\text{sind}(x)$ ، میتوان آنرا بر حسب درجه نمود.

تمرین (سعی کنید مطالب زیر را در پنجره فرمان تایپ کنید):

```
clc;
x = pi/2;
y = sin(x)

z = atan(1.0)
x = 30;
w = sind(x)
z = atand(1.0)
```

■ توابع نمائی، ریشه دوم، و خطای:

| | |
|--------|-----------------|
| exp | نمائی |
| log | لگاریتم طبیعی |
| log 10 | لگاریتم پایه ۱۰ |
| sqrt | جذر |
| erf | تابع خطای |

تمرین (سعی کنید مطالب زیر را در پنجره فرمان تایپ کنید):

```
clc;
x = 2.5;
y = exp(x)
z = log(y)
w = sqrt(x)
```

■ اعداد مختلط:

اعداد مختلط را می‌توان به دو صورت نوشت: یکی دستگاه کارتزین، مانند $z=x+jy$ ؛ یا دستگاه قطبی از قبیل $r \cdot \exp(j\theta)$. توجه کنید که در سراسر کتاب z را برای $\sqrt{-1}$ بکار می‌بریم. البته، متلب بما اجازه میدهد که z را برای $\sqrt{-1}$ نیز بکار برد. توجه: i و j نیز برای نام متغیرهای متلب که اغلب در حلقه‌ها بکار می‌روند مجاز هستند. برای دوری از اشتباه، برنامه‌هایی که حاوی

اعداد مختلط هستند، از نام متغیرهای i و j استفاده نکنید.

| | |
|-------|----------------------------|
| abs | absolute value (magnitude) |
| angle | phase angle (in radians) |
| conj | complex conjugate |
| imag | complex imaginary part |
| real | complex real part |

تمرین (سعی کنید مطالب زیر را در پنجره فرمان تایپ کنید):

```
clc;
z1 = 1 + j;
z2 = 2 * exp(j * pi/6)
y = abs(z1)
```

سایر توابع مفید:

تعداد عناصر بردار X را بما میدهد $\text{length}(X)$

اندازه (تعداد سطرها و تعداد ستونهای) ماتریس X را میدهد $\text{size}(X)$

برای بردارها، $\text{sum}(X)$ جمع عناصر در X را میدهد. برای ماتریس‌ها، یک بردار سطري شامل جمع عناصر در هر ستون ماتریس را میدهد.

برای بردارها، $\text{max}(X)$ عنصر بیشینه در X را میدهد. برای ماتریس‌ها، یک بردار سطري شامل عناصر بیشینه در هر ستون ماتریس را میدهد. اگر X یک بردار ستونی باشد، عنصر بیشینه X را میدهد.

مانند $\text{max}(X)$ با این تفاوت که عنصر کمینه در X مطرح است. $\text{min}(X)$

برای بردارها، $\text{sort}(X)$ جوهر کردن عناصر X بترتیب صعودی است. برای ماتریس‌ها، $\text{sort}(X)$ هر ستون در ماتریس را بترتیب صعودی مرتب می‌کند.

$$n! = 1 \times 2 \times 3 \times \cdots \times n \quad \text{factorial}(n)$$

عملگر مدول است، و باقی‌مانده نتیجه تقسیم x بر y را میدهد. برای مثال، $\text{mod}(x,y)$ یعنی $5 \div 13$ برابر 2 است بعلاوه باقیمانده 3 (در اینجا 2 در نظر گرفته نمی‌شود). مثالی دیگر، $\text{mod}(n,2)$ اگر n زوج باشد برابر صفر و اگر فرد باشد مساوی یک است.

تمرین (سعی کنید مطالب زیر را در پنجره فرمان تایپ کنید):

```
clc;
A = [2 15 6 18];
length(A)
y = max(A)
z = sum(A)
A = [2 15 6 18; 15 10 8 4; 10 6 12 3];
x = max(A)
y = sum(A)
size(A)
mod(21,2)
mod(20,2)
```

- فهرستی از مجموعه توابع اولیه ریاضیات را می‌توانید با تایپ کردن `help elfun` در پنجره فرمان مشاهده نمایید.
- بعضی وقت‌ها، لازم است اندازه یک ماتریس را تخصیص دهیم. این را می‌توان با تعریف یک ماتریس تمام صفر یا یک، مانند زیر، تعریف کرد:

```
0 0 0
A = zeros(3) = 0 0 0
0 0 0
0 0
B = zeros(3,2) = 0 0
0 0
1 1 1
C = ones(3) = 1 1 1
1 1 1
1 1 1
D = ones(2,3) = 1 1 1
1 1 1
```

تابعی که ماتریس همانی را تولید می‌کند (ماتریسی که فقط قطر اصلی آن یک و بقیه عناصر صفر هستند) با `eye` درست می‌شود. مثلاً:

```
1 0 0
I = eye(3) = 0 1 0
0 0 1
```

۵.۲ ورودی و خروجی متلب

■ اگر می‌خواهید برنامه در جایی متوقف و در نگی داشته تا از صفحه کلید ورودی دریافت کند، از تابع `input` استفاده کنید. برای مثال، برای وارد کردن ماتریس ۲ در ۳، بدین صورت عمل کنید

```
Z = input('Enter values for Z in brackets \n');
```

سپس این را تایپ کنید
`[5.1 6.3 2.5; 3.1 4.2 1.3]`

بنابراین،

$$Z = \begin{matrix} & 5.1 & 6.3 & 2.5 \\ & 3.1 & 4.2 & 1.3 \end{matrix}$$

توجه کنید که آرگومان `()` رشته مشخصه‌ای است که با دو پرانتز محصور گشته، که بر روی صفحه نمایشگر تایپ می‌شود. خط جدید `\n` به مطلب می‌گوید که مکان نما^{۱۷} را به خط بعدی حرکت بدهد. بر عکس، `\t` (تب - tab) به متلب می‌گوید که مکان نما به چندین فاصله در طول همان خط حرکت دهد.

اگر می‌خواهید داده متنی به ورودی `input` وارد کنید، لازم است متن را با یک پرانتز محصور کنید. البته، می‌توانید از این صرفنظر کرده و با آرگومان ثانوی `'s'` به ورودی `input` ورودی مطابق متن زیر عمل نمائید.

```
response = input('Plot function? (y/n):\n', 's');
```

در این حالت، کاربر می‌تواند با وارد کردن `y` یا `n` (بدون هیچگونه علامت دیگری) پاسخ گوید.

تمرین (سعی کنید مطالب زیر را در پنجره فرمان تایپ کنید):

```
z = input ('Enter a 2x3 matrix of your choosing\n')
name=input('Enter name enclosed by single quote marks:')
response = input('Plot function? (y/n):\n', 's');
```

■ برای فرمان چاپ `disp` درست محتوای ماتریس یا اطلاعات عددی؛ برای مثال (فرض کنید که `X` اکنون در پنجره فرمان وارد شده است):

```
>> x = [3.6 7.1]; disp(X); disp('volt');
```

^{۱۷}cursor

بنابراین در صفحه نمایشگر مطالب زیر نمایش داده خواهد شد:

3.6000 7.1000

volt

>>

■ فرمان چاپ `fprintf` متن فرمت شده (فرم و نحوه نوشتمن را فرمت می‌گوئیم) در صفحه نمایشگر یا در یک فایل؛ برای مثال:

```
>> I = 2.2;
>> fprintf ('The current is %f amps \n', I);
```

در روی صفحه نمایشگر این ظاهر می‌شود:

The current is 2.200000 amps

علامت `f` استناد به تعداد نقاط شناور فرمت بوده، و پیش فرض شش رقم اعشار است. (در متلب های قدیم `f` چهار رقم اعشار بود).

فرمان `fprintf` بر اساس زبان برنامه نویسی C تدوین شده است [۱]. بنابراین، شما می‌توانید کمترین فضا برای چاپ متغیرها و همین‌طور تعداد اعداد اعشاری را با استفاده از

`%8.2f`

این هشت فضا برای متغیرها با دو فضا برای اعشاری در نظر می‌گیرد. شما نیز می‌توانید مکان اعشار را مشخص کنید. مثلاً، بگوئید سه رقم اعشار، و متلب بطور خودکار فضای متغیر را تصمیم می‌گیرد

`.3f`

البته، برای نوشتمن جداولی تمیز و مرتب، بهتر است تعداد فضای را در فرمت مشخص کنید.

سایر فرمتهای چنین هستند:

`%i` یا `%d` برای اعداد صحیح بکار می‌رود

`%e` نمایش علمی (مانند `6.02e23`) پیش فرض شش رقم اعشار

`%g` بطور خودکار کمترین `%e` یا `%f` را بکار می‌برد

`%s` برای رشته‌ای کاراکتر استفاده می‌شود.

`%c` برای یک کاراکتر تکی استفاده می‌شود.

بر خلاف C، رشته فرمت در `fprintf` متلب باید با یک علامت نقل قول بسته شود.

بعضی وقت‌ها، فرمت بخشی از فرمان `fprintf` ممکن است خیلی بلند باشد که در یک سطر شما جای گیرد. وقتی این اتفاق افتاد، می‌توانید از سه نقطه (...) استفاده کرده و به متلب

بگوئید که در خط بعدی، مانند مثال زیر، آنرا ادامه دهد.

```
fprintf('resistance = %f volts = %f current = %f \n', ...
R,V,I);
```

سه نقطه را می‌توان برای ادامه هر بیان متلب در خط بعدی بکار برد تا متن را مطابق مثال زیر خواناتر نماید:

```
a = (-4*pi^2*omega^2*r^2*(cos(arg1))^2/sqrt(arg2) +...
4*pi^2*omega^2*r^2*(sin(arg1))^2/sqrt(arg2)) *...
exp(j*pi*omega);
```

یک روش دیگر برای استفاده نکردن از سه نقطه در توسعه طول یک عبارت ریاضی، تقسیم عبارت ریاضی به خطوط متعدد و سپس جمع کردن آنها است. بعنوان مثال، می‌توانیم رابطه فوق را به سه سطر کوچکتر تقسیم کرده و سپس a را محاسبه کنیم:

```
a1 = -4*pi^2*omega^2*r^2*(cos(arg1))^2/sqrt(arg2);
a2 = 4*pi^2*omega^2*r^2*(sin(arg1))^2/sqrt(arg2);
a = (a1+a2) * exp(j*pi*omega);
```

■ چاپ کردن یک فایل: غالباً میخواهیم که نتیجه برنامه متلب را برای گزارش و نتیجه‌گیری در فایلی چاپ و نگهداری کنیم. علاوه، خروجی برنامه که در فایلی چاپ شده است را می‌توان در درون فایل ویرایش کرده، بقسمی که آنها را بصورت ستونی در یک جدول ویرایش و قرار داد. قبل از آنکه بتوانید آنرا در فایلی چاپ کنید، لازم است برای چاپ کردن فایلی باز کنید که این کار را با دستور `fopen` انجام می‌شود. نحوه کار برای `fopen` چنین است:

```
fo = fopen('filename', 'w')
```

در این صورت `fo` در فایلی بنام `filename` بوده، و `w` عمل نوشتن را نشان میدهد.

برای چاپ کردن به `filename` این طور عمل کنید.

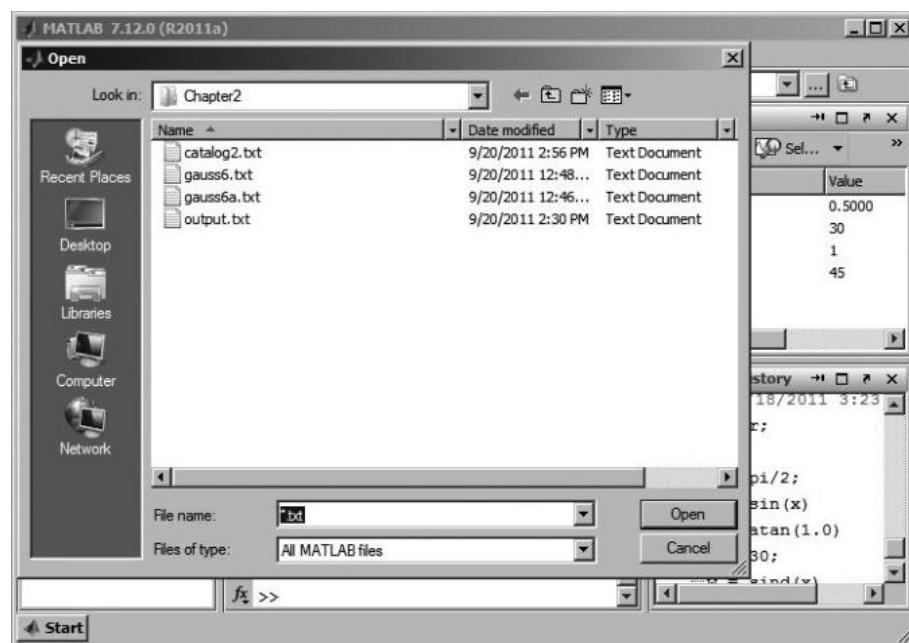
```
fprintf(fo, 'format', var1, var2, ...);
```

که فرمت شامل فرمت متن برای `var1`, `var2` و غیره است.

تمرین (این برنامه ساده را در پنجره متن نوشته و آنرا بصورت `io-example.m` ذخیره کنید):

```
% io_example.m
% This program is a test for printing to a file.
clear; clc;
V = 120;
I = 2.2;
fo = fopen('output.txt','w');
fprintf (fo, 'V = %4i volt, I = %5.2f amp \n',V,I);
fclose(fo);
```

پسوند در فایل خروجی باید `.txt` باشد. (در غیر این صورت، متلب با عبارت تعجب انگیز



شکل ۸.۲: فایلی با پسوند `.txt`

شروع می‌کند). نتیجه فایل خروجی می‌تواند با پنجره متنی متلب یا پنجره فرمان باز شود. برای دسترسی به فایل خروجی، روی `File-Open` کلیک کرده که صفحه شکل (۸.۲) را بالا می‌آورد. در جعبه‌ای که نام `File name` دارد، `*.txt` را تایپ کنید. این تمام فایل‌هایی که پسوند `.txt` دارند را برای شما می‌آورد. برای باز کردن فایل مورد نظر شما، دو بار روی نام فایل خروجی کلیک کنید (در این مثال، نام فایل `output.txt` است). در نسخه‌های قدیمی متلب، شما نمی‌توانستید فایل خروجی را بدون بیان `fclose(fo)` در برنامه باز کنید. اما بهتر است همواره در برنامه خود بعداز تمام عبارات خروجی کلمه `fclose` را نیز بنویسید.

■ فرمان `fscanf` را می‌توان برای خواندن از یک فایل موجود بصورت زیر بکار برد:

```
A = zeros(n,m);
fi = fopen('filename.txt', 'r');
[A] = fscanf(fi, '%f', [n,m]);
```

که در آن $n \times m$ تعداد عناصر در فایل داده است. کلمه 'r' در عبارت `fopen` نمایشگر این است که فایل برای خواندن داده در برنامه است.

ماتریس $n \times m$ بترتیب ستونی پر شده است. بنابراین، سطرها به ستون‌ها، و ستون‌ها به سطرها تبدیل می‌شود. برای استفاده از داده‌ها بترتیب اصلی خود، آنچه خوانده شده در ماتریس را ترانهاد^{۱۸} کنید. برای ترانهاده کردن ماتریس A در متلب، فقط لازم است 'A' را بنویسید. این باعث می‌شود جای سطرها و ستون‌ها عوض شود.

■ فایل داده‌ها را می‌توان با فرمان `load` بدرودن برنامه وارد کرد. فرمان `load`، بر عکس `fscanf` سطرها را سطر و ستون‌ها را ستون وارد می‌کند. برای مثال:

```
load filename.txt
x = filename(:,1);
y = filename(:,2);
```

فایل ورودی باید همان تعداد سطر را در هر ستون داشته باشد.

۶.۲ روند برنامه متلب

■ حلقه `for` وسیله‌ای برای انجام یک سری عملیات فقط با چند خط برنامه است. دستورالعمل آن چنین است:

```
for m = 1:20
    statement;
    :
    statement;
end
```

متلب مجموعه‌ای از عملیات را بین `for` و `end` را با مقدار ابتدا `m` برابر یک انجام داده و سپس آنرا برابر دو و همین طور تا `m` برابر 20 انجام میدهد. بعداز 20 بار تکرار، برنامه از حلقه خارج می‌شود. هر عبارتی که نباید تکرار شود، نبایستی درون حلقه `for` قرار داده شود. برای مثال، عنوان جداول که نباید تکرار شود باید خارج از حلقه `for` باشد.

^{۱۸}transpose

تمرین (این برنامه ساده را در پنجره متن نوشه و آنرا بصورت `for-loop-example.m` ذخیره کرده و اجرا نمایید):

```
% for_loop_example.m
clear; clc;
% Table headings:
fprintf('n      y1      y2 \n');
fprintf('-----\n');
% Using the index in the for loop in an arithmetic statement
for j = 1:10
    y1 = j^2/10;
    y2 = j^3/100;
    fprintf('%5i  %10.3f %10.3f \n',j,y1,y2);
end
```

توجه: اگر اندیس در حلقه `for` برای انتخاب عنصری از یک ماتریس بکار رود، در این صورت اندیس باید عدد صحیح باشد. البته، اگر از اندیس `for` برای اندیس ماتریس نخواهید استفاده کنید، در آن صورت اندیس `for` لزومی ندارد عدد صحیح باشد. و این در مثال زیر نشان داده شده است.

تمرین (این برنامه ساده را در پنجره متن نوشه و آنرا به نام `for-loop-example2.m` ذخیره کرده و اجرا نمایید):

```
% for_loop_example2.m
% Using a range of non-integer x values in a for loop.
% The range of x is from -0.9 to +0.9 in steps of 0.1.
clear; clc;
% print the table header outside of the 'for'loop:
fprintf('x      y1      y2 \n');
fprintf('-----\n');
for x = -0.9:0.1:0.9
    y1 = x/(1-x);
    y2 = y1^2;
    fprintf('%5.2f  %10.3f %10.3f \n',x,y1,y2);
end
```

حلقه while ■

دستورالعمل:

```
n = 0;
while n < 10
    n = n+1;
    statement;
    :
    statement;
end
```

در حلقه while، متلب عملیات بین while و end را تا وقتی که شرط ارضاء شود انجام و تکرار می‌شود. اگر یک اندیس در برنامه لازم باشد، استفاده از حلقه while (برخلاف حلقه for) لازم است که برنامه خودش یک اندیس، مانند آنچه در بالا نشان داده شده است، درست کند. توجه کنید که $n=n+1$ ^{۱۹} ممکن است از نظر جبری بی معنی باشد، اما در زبان متلب مفهومی در بر دارد. عملگر " = " در متلب (مانند تمام زبانهای کامپیوتری) عملگر واگذاری^{۲۰} است، بدین معنی که به متلب می‌گوید برو و محتوای حافظه‌ای که شامل متغیر n است را بیاور، مقدار آنرا در واحد محاسباتی CPU (واحد پردازش مرکزی^{۲۰}) قرار بده، یک واحد به آن اضافه کن، و دوباره مقدار جدید را در همان حافظه‌ای که برای n اختصاص داشت، قرار بده. بنابراین، مقدار قدیم n با مقدار جدید n عوض می‌شود.

دستور if ■

دستورالعمل:

```
if      جملات منطقی
        statement;
        :
        statement;
else
        statement;
        :
        statement;
end
```

اگر جمله منطقی درست باشد، در این صورت فقط ب عبارت‌های بالائی محاسبه می‌شود. اگر جمله منطقی غلط باشد، فقط عبارت‌های پائینی محاسبه می‌شود.

^{۱۹}assignment^{۲۰}central processing unit

■ جملات منطقی بفرم های زیر هستند

| | |
|------------------------|---|
| <code>a == b;</code> | <code>a <= b;</code> |
| <code>a < b;</code> | <code>a >= b;</code> |
| <code>a > b;</code> | <code>a ~= b; (a not equal to b)</code> |

■ جملات منطقی مرکب

| | |
|---|-------------------------------------|
| <code>a > b && a ~= c</code> | <code>(a > b AND a ≠ c)</code> |
| <code>a > b a < c</code> | <code>(a > b OR a < c)</code> |

■ if-elseif نردنیان

دستور العمل:

```

if logical expression 1
    statement(s);
elseif logical expression 2
    statements(s);
elseif logical expression 3
    statement(s);
else
    statement(s);
end

```

نردنیان if-elseif از بالا به پائین اجرا می شود. اگر جمله منطقی بالائی درست باشد، عبارت مربوط به آن جمله منطقی محاسبه شده، و برنامه نردنیان را ترک می کند. اگر جمله منطقی بالائی غلط باشد، برنامه به جمله منطقی بعدی می روید. اگر آن هم صحیح باشد، برنامه آن سری عبارات مربوط به آن را اجرا کرده و نردنیان را ترک می کند. اگر جمله منطقی غلط باشد، برنامه به جمله منطقی بعدی می روید و این عمل ادامه می یابد. اگر هیچ از عبارات صحیح نباشند، برنامه عبارت مربوط به elseif را اجرا می کند. البته احتیاجی به عبارت elseif الزامی نیست. در این حالت، اگر هیچ یک از جملات منطقی صحیح نباشند، هیچ یک از عبارات های درون نردنیان اجرا نمی شود.

■ فرمان break را می توان درون حلقه for یا while برای خاتمه دادن حلقه اجرائی بکار برد.

برای مثال:

```
for m = 1:20
    statement;
    :
    statement;
    if m > 10
        break;
    end
end
```

در این مثال، وقتی `m` بیشتر از ۱۰ می‌شود، برنامه حلقه `for` را ترک کرده و به عبارت بعدی برنامه می‌پردازد.

■ گروه سویچ switch

دستور العمل:

```
switch(var)
    case var1
        statement(s);
    case var2
        statement(s);
    case var3
        statement(s);
    otherwise
        statement(s);
end
```

که در آن `var` بر حسب مقادیر ممکن `var1`, `var2`, `var3`، وغیره عبارتی را اجرا می‌کند.
اگر `var` برابر `var1` باشد، عبارت مربوط به `var1` اجرا شده، و برنامه از گروه سویچ خارج می‌شود.
اگر `var` برابر `var1` نباشد، برنامه `var` را برای برابر بودن `var2` امتحان می‌کند و اگر جواب مثبت بود، برنامه عبارت مربوط به `var2` را انجام داده و گروه سویچ را ترک می‌نماید. اگر `var` برابر هیچ یک از `var1`, `var2` وغیره نباشد، برنامه عبارت مربوط به `otherwise` را اجرا می‌کند. اگر `var1`, `var2` وغیره رشته باشند، آنها را با علامت نقل قول محدود کرد. باستی توجه کرد که `var` نمی‌تواند عبارت منطقی، از قبیل `var1>=80` باشد.

تمرین (این برنامه ساده را در پنجره متن نوشته و آنرا بنام `switch-example.m` ذخیره کرده و اجرا نمائید):

```
% switch_example.m
% This program is a test of the switch statement.
clear; clc;
var = 'a';
```

```

x = 5;
switch(var)
    case 'b'
        z = x^2;
    case 'a'
        z = x^3;
end
fprintf('z = %6.1f \n',z);

```

برنامه‌های تمرینی اضافی که نشانده‌نده استفاده از حلقه `for`، حلقه‌های `while` تو در تو، حلقه عبارت `if`، `break`، `input`، `elseif` در زیر آورده شده است. چون تعداد این برنامه‌های نمونه شامل تعیین e^x توسط بسط سری تیلور^{۲۱} است، ما با بحث بسط سری e^x شروع می‌کنیم.

مثال ۱.۲: محاسبه e^x

| Term index: | 1 | 2 | 3 | 4 | ... | n |
|-------------|---|---|---|---|-----|-----|
|-------------|---|---|---|---|-----|-----|

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots + \frac{x^n}{n!}$$

چندین الگوریتم برنامه نویسی برای محاسبه این سری وجود دارد. ساده‌ترین روش شروع با `sum=1` و سپس محاسبه هر جمله با استفاده از توان وتابع فاکتوریل مطلب و جمع جمله از `sum` است. رویکرد دیگر محاسبه هر جمله تک تک، ذخیر آنها در یک رشته، و سپس استفاده از تابع جمع مطلب برای جمع کردن تمام آنها است. البته، برای برخی سری‌ها بهتر است هر جمله از سری را بر مبنای مقدار جمله قبلی محاسبه کرد. برای مثال، در سری بالا، میتوانیم ببینیم که جمله سوم در سری را می‌توان از جمله دوم با ضرب جمله دوم در x و تقسیم بر سه؛ یعنی $\times x/3$ جمله دوم = جمله سوم بدست آورد. بطور کلی $\times x/n$ $(n-1)$ جمله = (n) جمله.

```

% Example_2_1.m
% This program calculates e^x by series and by MATLAB's
% exp() function. The 'for' loop ends when the term only
% affects the seventh significant figure.
% Note: e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + ...

```

^{۲۱}Taylor series expansion

```

clear; clc;
x=5.0;
s=1.0;
for n=1:100
    term=x^n/factorial(n);
    s=s+term;
    if(abs(term) <= s*1.0e-7)
        break;
    end
end
ex1=s;
ex2=exp(x);
fprintf('x=%3.1f ex1=%8.5f ex2=%8.5f \n',x,ex1,ex2);
%-----

```

مثال ۲.۲: حلقه‌های تو در تو

```

% Example_2_2.m
%
% This program illustrates the use of nested loops, i.e.
% an inner 'for' loop inside an outer 'for' loop.
%
% The program calculates  $e^x$  by both MATLAB's 'exp'
% command (variable 'ex2'), and by a Taylor series
% expansion (variable 'ex1'), where  $-0.5 < x < 0.5$ .
%
% The outer 'for' loop is used to determine the x
% values. The inner loop is used to determine the Taylor
% series method for evaluating  $e^x$ . In this example,
% term(n+1) is obtained by multiplying term(n) by  $x/n$ .
%
% The variable 'term' is established as a vector so that
% MATLAB's built-in 'sum' function can be used to sum
% all the terms calculated in the Taylor series method.
%
% A maximum of fifty terms is used in the series.
%
% Program output is sent both to the screen and to a
% file. By printing the output to a file, you can easily

```

```
% edit the output file to line up column headings,
% etc. (which you can't do when printing to the screen).
% Note: e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + ...
clear; clc;
xmin=-0.5; dx=0.1;
fo=fopen('output.txt','w');
% Table headings
fprintf(' x           ex1           ex2 \n');
fprintf('-----\n');
fprintf(fo,' x           ex1           ex2 \n');
fprintf(fo,'-----\n');
for i=1:11
    x=xmin+(i-1)*dx;
    ex2=exp(x);
    term(1)=1.0;
    for n=1:49
        term(n+1)=term(n)*x/n;
        if abs(term(n+1)) <= 1.0e-7
            break;
        end
    end
    ex1 = sum(term);
    fprintf('%5.2f %10.5f %10.5f \n',x,ex1,ex2);
    fprintf(fo,'%5.2f %10.5f %10.5f \n',x,ex1,ex2);
end
fclose(fo);
%-----
```

توجه: در این مثال، دو عبارت

```
for i=1:11
    x=xmin+(i-1)*dx;
```

را می‌توان با یک عبارت زیر عوض کرد

```
for x= -0.5:0.1:0.5
```

این رویکرد وقتی اندیس حلقه `for` بکار برد نمی‌شود تا عنصری را در یک ماتریس انتخاب کند، قابل قبول است. در متلب، انتخاب متغیر برای عناصر یک ماتریس باید عدد صحیح باشد.

مثال ۳.۲: حلقه while

```
% Example_2_3.m

% Calculation of e^x by both MATLAB's exp function and
% the Taylor series expansion of e^x. The input()
% function is used to establish the exponent x.

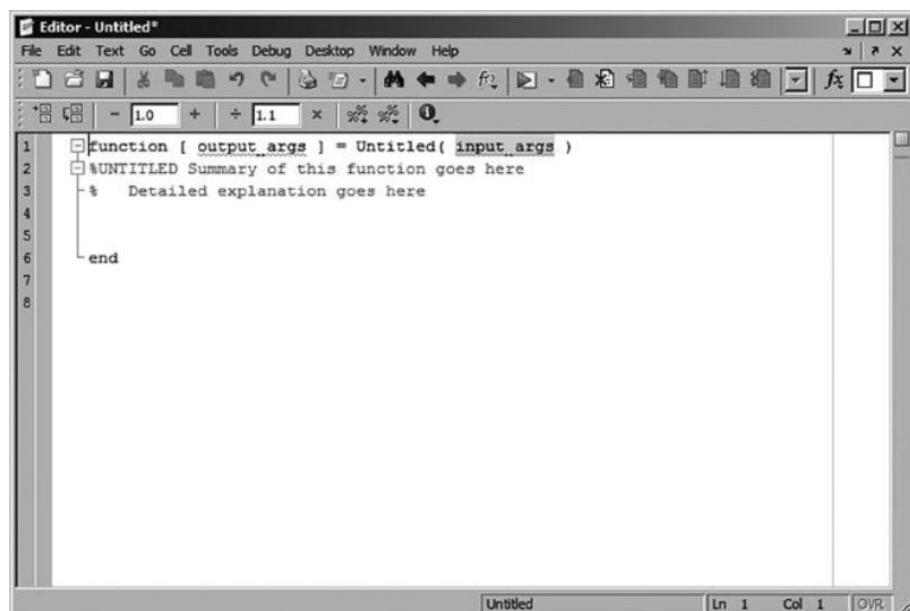
% A 'while' loop is used in determining the series
% solution. A 'break' statement is used to end the loop
% if the number of terms becomes greater than 50. The
% display statement is used to display the values of x,
% ex and ex2 in the command window.

% Note: e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + ...
clear; clc;

x=input('Enter a value for the exponent x \n');
n=1; ex=1.0; term=1.0;
while abs(term) > ex*1.0e-6
    term=x^n/factorial(n);
    ex=ex+term;
    n=n+1;
    if n > 50
        break;
    end
end
ex2=exp(x);
disp(x);
disp(ex);
%-----
```

۷.۲ فایل‌های توابع متلب

اگر برنامه پیچیده‌ای دارید و می‌خواهید آنرا به بخش‌های کوچکتری تقسیم کنید، توابع متلب مفید خواهد بود. همچنین، اگر یک سری عبارت می‌بایست دفعات متعدد بکار رود، ساده‌تر آن است که آنرا بصورت تابع در آورید. تابع معادل زیربرنامه^{۲۲} در اکثر زبان‌های برنامه‌نویسی است، اما در متلب عموماً آنها را در فایل جداگانه در برنامه اصلی خود قرار می‌دهند. (ضمناً تابع کوچک را نیز می‌توان در همان متن برنامه اصلی نوشت. این موضوع در بخش بعدی بحث می‌شود.)



شکل ۹.۲: الگوی تابع

■ نام فایل تابع باید بصورت `functioname.m` ذخیره شود.

■ متلب دارای الگویی برای نوشتن یک تابع دارد (به شکل ۹.۲ مراجعه کنید). اولین عبارت قابل محاسبه در فایل تابع باید `function` باشد. بطوری که در شکل (۹.۲) مشاهده می‌شود، الگوی تابع بصورت زیر است:

```
function [output arguments] = Untitled(input arguments)
    % این بخش تعاریف توابع در جدول (۱.۲) نشان داده شده است.
```

اگر تابعی دارای بیش از یک مقادیر خروجی باشد، در این صورت متغیرهای خروجی باید در داخل براکت قرارداده شود. اگر فقط یکی مقدار خروجی باشد، دیگر براکت لازم نیست. اگر هیچ

^{۲۲} subroutine

جدول ۱.۲: نمونه‌هایی از کاربرد توابع

| <i>Function Definition Line</i> | <i>Function File Name</i> |
|---------------------------------|---------------------------|
| function [P,V] = power(i,v) | <i>power.m</i> |
| function ex = exf(x) | <i>exf.m</i> |
| function [] = output(x,y) | <i>output.m</i> |

خروجی نخواهیم، برآکت خالی استفاده می‌کنیم.

- تفاوت یک تابع از یک متن در این است که آرگومانها را می‌توان بیک تابع دیگر یا پنجره فرمان برگردانید. متغیرهای تعریف شده و محاسبه شده درون تابع در تابع جای دارند.
- در خیلی از متلب‌ها توابع " built-in " بصورت ام فایل *m*. درست شده است. برای مثال، فاکتوریل *factorial* بصورت فایل *factorial.m* درست شده است. شما می‌توانید با تایپ کردن *which factorial* در پنجره فرمان *factorial.m* را پیدا کنید.

مثال ۴.۲: تعریف تابع متلب

```
%  
% exf1.m (Example 2.4)  
% Define a function, 'exf1', that evaluates e^x.  
% This function takes 'x' as an input argument and is  
% called either from the Command Window or from another  
% program. The resulting output, 'ex', is available to  
% be used in another program or in the Command Window.  
% In this example, term(n+1) is obtained from term(n)  
% by multiplying term(n) by x/n.  
% Note: e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + ...  
function ex = exf1(x)  
term(1)=1.0;  
for n=1:99  
    term(n+1)=term(n)*x/n;  
    if abs(term(n+1)) <= 1.0e-7
```

```

break;
end
end
ex=sum(term);
%-----

```

برای آزمایش کردن این تابع، آنرا در پنجره فرمان اجرا کنید. برخی از مثال‌ها در زیر آورده شده است:

```

>> exf1(1.0)
ans =
    2.7183
>> y = exf1(5.0)
y =
    148.4132

```

مثال ۲.۵: تعریف تابع متلب

```

%
% Example_2_5.m
% This program uses the function exf1 (defined in
% Example 2.4) in several arithmetic statements, i.e to
% calculate w, y and z. In this program, the output is
% sent to a file.
clear; clc;
fo=fopen('output.txt','w');
fprintf(fo,' x           y           w           z   \n');
fprintf(fo,'----- \n');
for x=0.1:0.1:2.0
    y=exf1(x);
    w=5*x*exf1(x);
    z=10*x/y;
    fprintf(fo,'%.4.2f  %.8.3f  %.8.3f  %.8.3f \n',x,y,w,z);
end
fclose(fo);
%-----

```

تمرین‌های زیر نشان میدهد که نام آرگومان‌ها در خواندن برنامه لازم نیست که همان نامی باشد که در تابع است. فقط فهرست آرگومان‌ها در برنامه خواندن لازم است که بهمان صورت ردیف آرگومان تابع تعریف شده باشد. این صورت وقتی یک تابع باید با چندین متن مختلف استفاده شود مفید‌تر خواهد بود. هر متن نام متغیر مختلف اما با هر نام متغیر مربوط به متغیر تابع دارد.

مثال ۶.۲: تعریف و استفاده تابع متلب

```
% Example_2_6.m
% This program uses the function exf2 to calculate
% q=e^w/z . w and z are input variables to function
% exf2. The function exf2 calculates q and returns it
% to this program. The input variables to function exf2
% need not be the same names as those used in the
% calling program. For one-to-one correspondence, the
% argument positions in the calling program have to be
% the same as the argument positions in the function.
% Note: function exf2 (defined below) must be created
% before this program is executed because the function
% exf2 calculates exy and returns it to the calling program.
% q is set equal to exy.

clear; clc;
z=5;
for w=0.1:0.1:2.0
    q=exf2(w,z);
    fprintf('Values of w,z and q from main program: ');
    fprintf('w=% .2f z=% .0f q=% .5f \n', w, z, q);
end
%-----
% exf2.m
% This script is used with script Example_2_6.m and creates a
% function 'exf2' that evaluates e^x/y. In this
% function, x and y are input variables that need to be
% defined in another program. The output variable in the
```

```
% function exf2 is sent back to the calling program.  
% This example demonstrates that the names of the  
% arguments in the calling program need not be the same  
% as those in the function. It is the order list in the  
% calling program that needs to be the same as the order  
% list in the function.  
  
% In this example, term(n+1) is obtained from term(n) by  
% multiplying term(n) by x and dividing by index n. The  
% program does its own summing of terms.  
  
% Note:  $e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + \dots$   
  
function exy = exf2(x,y)  
  
% Note: x and y are set equal to w and z respectively.  
s=1.0; term=1.0;  
  
for n=1:100  
    term=term*x/n;  
    s=s+term;  
    if abs(term) <= s*1.0e-5  
        break;  
    end  
end  
  
exy=s/y;  
  
% These fprintf statements demonstrate the equivalence  
% between function input variables (x,y) and calling  
% program variables (w,z) in the program Example_2_6.m.  
% Compare the printout of x,y and exy with w,z and q.  
fprintf('Values of x,y and exy from exf2():      ');\nfprintf('x=% .2f y=% .0f exy=% .5f \n', x, y, exy);  
%-----
```

۸.۲ توابع بی‌نام

بعضی وقت‌ها، راحت‌تر است که یک تابع را بجای فایل جداگانه در داخل متن تعریف کنیم. برای مثال، اگر یک تابع خلاصه (شاید هم یک خط) باشد، و برعکس بخواهیم در متن استفاده کنیم، در آن صورت شکل بی‌نام تابع را استفاده می‌توان کرد. این از ذخیره کردن یک ام‌فایل دیگر برای شما جلوگیری می‌کند. مثلاً:

```
fh = @(x,y)(x*\sin(x)+x*\cos(y));
```

متلب علامت @ تابع دستی^{۲۳} تعریف می‌کند. یک تابع دستی معادل دسترسی به تابع با خواندن نام تابع بوده که در این حالت برابر `fh` است. عبارت (x,y) آرگونهای ورودی برای تابع تعریف کرده و $((y)*\sin(x)+x*\cos(y))$ تابع واقعی است. توابع بی‌نام را می‌توان در متن یا پنجره فرمان بکار برد. برای مثال، در پنجره فرمان، عبارات زیر را تایپ کنید:

```
>> fh = @(x,y)(y*sin(x)+x*cos(y));
>> w = fh(pi,2*pi)
w =
3.1416
%-----
```

اطلاعات اضافی مربوط به توابع بی‌نام را می‌توانید با تایپ کردن `help function-handle` در پنجره فرمان بدست آورید.

۹.۲ گرافیک متلب

■ فرامین رسم:

| | |
|----------------------------|--|
| <code>plot(x,y)</code> | رسم خطی y در مقابل x |
| <code>semilogx(x,y)</code> | رسم نیمه‌لگاریتمی (مقیاس لگاریتمی برای x مقیاس خطی برای y) |
| <code>semilogy(x,y)</code> | رسم نیمه‌لگاریتمی (مقیاس خطی برای x مقیاس لگاریتمی برای y) |
| <code>loglog(x,y)</code> | رسم لگاریتمی-لگاریتمی (مقیاس لگاریتمی برای هر دو x و y) |
| <code>plot(x,y,w,z)</code> | رسم چندین منحنی روی یک محور: در این حالت y در مقابل x و z در مقابل w . برای مثال، فرض کنید |

$$A = \begin{matrix} t_1 & y_1 & z_1 & w_1 \\ t_2 & y_2 & z_2 & w_2 \\ \vdots & \vdots & \vdots & \vdots \\ t_n & y_n & z_n & w_n \end{matrix}$$

اجازه دهید $W = A(:,4)$ و $Z = A(:,3)$ ، $Y = A(:,2)$ ، $T = A(:,1)$ داشت

$$T = \begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{matrix} \quad Y = \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{matrix} \quad Z = \begin{matrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{matrix} \quad W = \begin{matrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{matrix}$$

^{۲۳}function handle

روشهای عددی و تحلیلی برای مهندسین برق با متلب

در این صورت، برای رسم Y در مقابل T , Z در مقابل T , و W در مقابل T تماماً در روی یک شکل، بنویسید:

```
plot(T, Y, T, Z, T, W);
```

البته، میتوانستیم از اقدامات اضافی خودداری کنیم و بنویسیم

```
plot(A(:,1), A(:,2), A(:,1), A(:,3), A(:,1), A(:,4));
```

■ رسمیت دادن به ترسیم‌ها:

میتوانید به رسم خود با فرمان grid آنرا شبکه‌بندی کنید.
برای محورهای x ها و y ها میتوانید برچسب گذاشته و عنوان برای رسم خود اضافه کنید. این فرامین را در برنامه خود اضافه کنید:

```
xlabel('T');
ylabel('Y,Z,W');
title('Y,Z,W vs T');
```

ضمناً، میتوانید به رسم خود متنی با فرمان زیر اضافه کنید

```
text(x position, y position, 'text statement');
```

حروف یونانی و علائم ریاضی را می‌توانید در برچسب xlabel, ylabel, title و با text نوشتن الفبائی نام حروف یونانی و قبل از آن از علامت '\' اضافه کنید. مثلًا، برای نمایش ω ، چنین بنویسید \omega و برای نمایش γ بنویسید \gamma. برای مثال:

```
ylabel('\omega'), title('\omega vs. \gamma'), text(10,5,'Phi');
```

برای فهرست کامل علائم یونانی و کاراکترهای خاص، به پیوست ب مراجعه کنید. بعضی وقت‌ها لازم دارد علامت "" در روی برچسب محورها یا عنوان ترسیم خود چاپ کنید. در این حالت، از دو تا ' برای فرار از یک کاراکتر رشتہ خود استفاده کنید. بنابراین، برای تولید عنوان رسم "سیگنال'A' در مقابل سیگنال'B'"، چنین تایپ کنید.

```
title('Signal ''A''vs. Signal ''B'')
```

در اکثر حالتها، اجازه دهد متلب مقیاس را برای محورهای x و y رسم دو بعدی شما انتخاب کند. البته، اگر از انتخاب خودکار متلب راضی نشیدید، در آن صورت می‌توانید از فرمان axis برای تنظیم حدود رسم منحنی‌های خود استفاده کنید:

```
axis([xmin xmax ymin ymax])
```

که در آن xmin و xmax مقیاس محور x ها و ymin و ymax مقیاس محور y ها را مشخص می‌کند.

■ تعامل ترسیم‌ها:

بعنوان یک روش دیگر برای اضافه کردن فرامین xlabel, ylabel، و title به برنامه خود، می‌توانید نمودارهای را رسم کرده و سپس در پنجره نمودار، روی منو Insert کلیک کرده و با پائین کشیدن منو X Label را انتخاب کنید. این یک جعبه‌ای را مشخص می‌کند که شما می‌توانید نام متغیر افقی را تایپ کنید. می‌توانید این را برای Y Label و Title نمودار خود

انجام دهید. سایر گزینه‌ها در منو Insert برای TextBox، Text Arrow، Arrow وجود دارد. وقتی روی هر کدام از این گزینه‌ها کلیک کنید، یک علامت بعلاوه می‌آید که شما می‌توانید با حرکت موس هر کجا بخواهید روی نمودار آنرا قرار داده و مطالب مورد نظر خود را درج نمائید. برای حذف TextBox، نشانک را در TextBox گذاشته و با موس راست کلیک کنید. این باعث می‌شود که منو پائین آورده شود؛ LineStyle را انتخاب کرده و سپس روی none چپ کلیک کنید.

■ ذخیره کردن ترسیم‌ها:

برای ذخیره کردن نمودار، روی File در پنجره نمودار کلیک کرده و گزینه Save از منو پائین آورده شده را انتخاب نمائید. این یک پنجره برای شما تولید می‌کند که شما می‌توانید نام فایل خود را وارد کنید. از معايیت این روش این است که اگر خواستید به متن برگردید، اقلامی که شما دستی وارد کرده‌اید ذخیره نخواهد شد. اگر خواستید، شکلی را برای گزارش خود کپی کنید، می‌توانید روی Edit در پنجره نمودار کلیک کرده، و سپس از منو پائین آورده شده Copy Figure را انتخاب کنید. در این صورت می‌توانید شکل را روی گزارش خود بچسبانید. اگر نمودار تک رنگ لازم دارید، (برای فتوگپی با بهترین کیفیت)، می‌توانید تمام منحنی‌های خود را با انتخاب منو فایل سیاه انتخاب کرده و Exort Setup را از منو پائین آورده شده انتخاب کنید. این پنجره‌ای باز خواهد کرد که شما لازم دارید. روی Rendering کلیک کرده و Colorspace را به black and white تغییر دهید.

گزینه‌های زیادتری نیز در پنجره نمودار هست؛ البته، این را به دانشجویان واگذار می‌کنیم که آنها را کاوش و جستجو کنند.

مثال ۷.۲: رسم نمودارها

```
% Example_2_7.m
% This program creates a simple table and a simple plot.
% First, the vectors y1, y2 and t are calculated.
% Next a table of y1 = n^2/10 and y2 = n^3/100 is created.
% Then y1 and y2 are plotted.
clear; clc;
for n=1:10
    t(n)=n;
    y1(n)=n^2/10;
    y2(n)=n^3/100;
end
% By making t, y1 and y2 as vectors, their values can be
% printed out outside the 'for' loop that created them.
% Print column headings
fprintf('      t          y1          y2      \n');
fprintf('-----\n');
for n=1:10
    fprintf('%8.1f    %10.2f    %10.2f \n', ...
            t(n),y1(n),y2(n));
end
% Create the plot. y1 is red, y2 is in green.
plot(t,y1,'r',t,y2,'g');
```

```

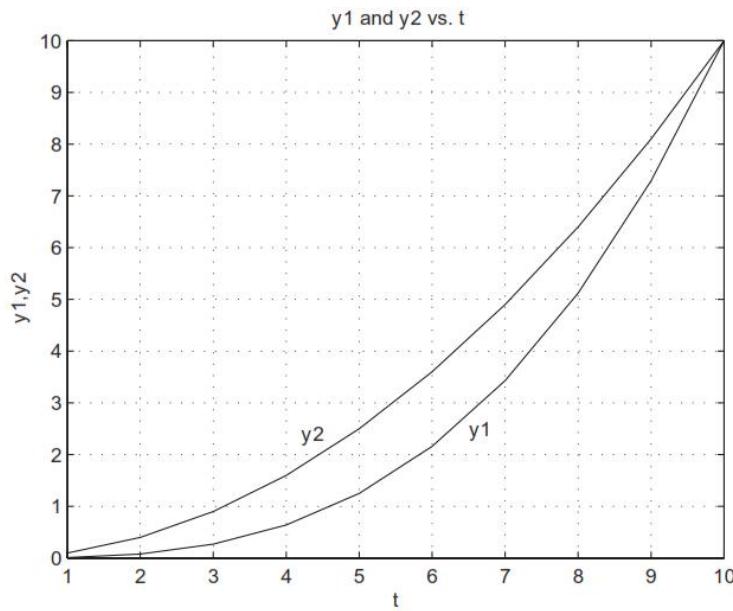
xlabel('t'), ylabel('y1,y2'), grid;
title('y1 and y2 vs. t');
% Plot identification is also established by adding text
% to the plot.
text(6.5,2.5,'y1');
% In the above statement, 6.5 is the abscissa position
% and 2.5 is the ordinate position where the 'y1' label
% will be positioned.
text(4.2,2.4,'y2');
% Other MATLAB plot options:
% Available color types:
%   blue          'b'
%   green         'g'
%   red           'r'
%   cyan          'c'
%   yellow        'y'
% Curves y1 and y2 can also be distinguished by using
% different types of lines as listed below.
% Available line types:
%   solid          (default)
%   dashed         '--'
%   dashed-dot    '-.'
%   dotted         ':'
% Alternatively, you can create a marker plot of
% discrete points (without a line) by using one of these
% marker styles:
%   point          '.'
%   plus           '+'
%   star           '*'
%   circle         'o'
%   x-mark         'x'
% The 'legend' command may also be used in place of the
% text command to identify the curves. The legend box
% may be moved by clicking on the box and dragging it to
% the desired position.
%-----

```

مثال ۸.۲: فرمان زیرنمودار subplot

فرض کنید می خواهید هر یک منحنی های مثال گذشته را در نمودار جداگانه ای، اما همه در یک صفحه رسم کنید. فرمان subplot این کار را انجام می دهد. فرمان subplot(m, n, p) صفحه را به ماتریس m در n نمودار تقسیم کرده و p موقعیت ماتریس نمودار را انتخاب می نماید. مثال زیر فرمان subplot را نمایش می دهد.

```
% Example_2_8.m
```



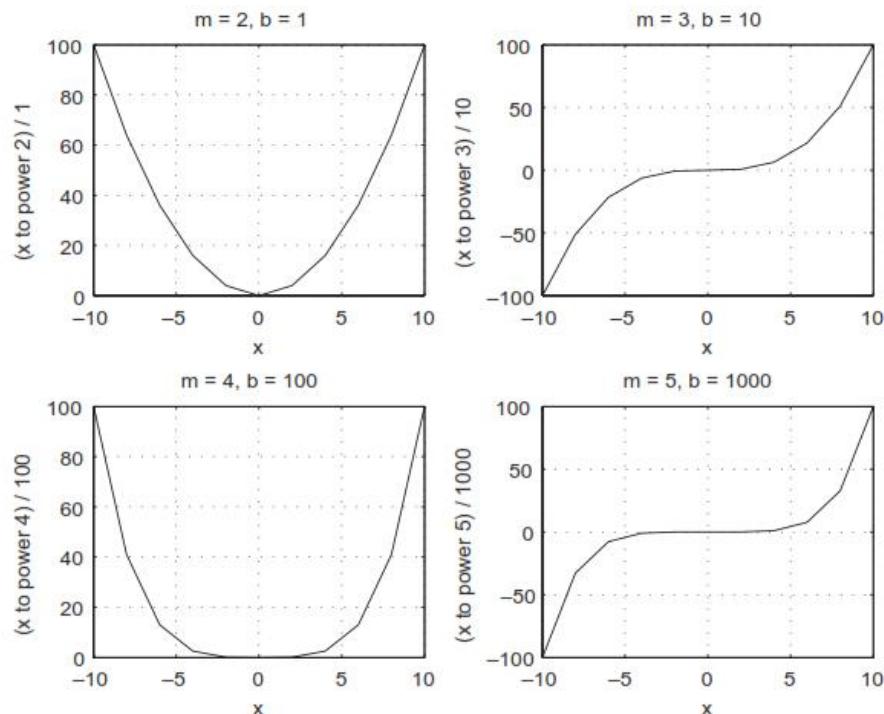
شکل ۱۰.۲: نمونه‌ای از یک نمودار که توسط فرمان plot تولید شده است.

```
% This program is an example of the use of the subplot
% command and the elseif ladder. Values of y1, y2, y3
% and y4 are constructed as vectors. Separate plots of
% y1 vs. t, y2 vs. t and y3 vs. t are plotted on the
% same page.
clc; clear;
for n=1:11
    t(n)=n-1;
    y1(n)=t(n)^2/10;
    y2(n)=sin(pi*t(n)/10);
    y3(n)=exp(t(n)/2);
    y4(n)=sqrt(t(n));
end
for n=1:4
    subplot(2,2,n)
    if n==1
        plot(t,y1), grid, title('y1 vs. t');
        xlabel('t'), ylabel('y1');
    elseif n==2
        plot(t,y2), grid, title('y2 vs. t');
        xlabel('t'), ylabel('y2');
    elseif n==3
        plot(t,y3), grid, title('y3 vs. t');
        xlabel('t'), ylabel('y3');
    elseif n==4
        plot(t,y4), grid, title('y4 vs. t');
        xlabel('t'), ylabel('y4');
    end
end
```

```

elseif n==4
    plot(t,y4), grid, title('y4 vs. t');
    xlabel('t'), ylabel('y4');
end
%-----

```



شکل ۱۱.۲: نمونه‌ای از زیرنمودار و استفاده از فرمان `sprintf` در عنوان نمودار و برچسب‌ها.

مثال ۹.۲: تابع `sprintf`

```

% Example_2_9.m
% This program demonstrates the use of sprintf command
% to pass formatted data to ylabel and title in a graph.
% The subplot command is also demonstrated.
clear; clc;
% Define the data to plot
x = -10:2:10;
m = 2:5;
b = [1 10 100 1000];
% Create table and plot
for i=1:4
    fprintf('m = %i \n',m(i));

```

```
% print table headings
fprintf('x yc \n');
fprintf('-----\n');
% (x to power m(i)) divided by b(i)
for j=1:length(x)
    yc(j) = x(j)^ m(i) / b(i);
    fprintf ('%5.0f %8.2f \n',x(j),yc(j));
end
fprintf ('\n\n');
% plot
subplot(2,2,i);
plot(x,yc);
xlabel('x');
ylabel(sprintf('(x to power %d) / %.0f',m(i),b(i)));
grid, title(sprintf('m = %d, b = %d',m(i),b(i)));
end
%-----
```

مثال ۱۰.۲: نمودار میخی

در پردازش سیگنالهای زمان گسسته، اغلب از نمودار میخی^{۲۴} برای نمایش رشته سیگنال استفاده می‌شود. فرض کنید سیگنال $x[n]$ برای مقادیر صحیح n داده شده است. فرمان $stem(n,x)$ تابع گسسته x را بصورت "آبنبات چوبی" عمودی که روی محور افقی هستند رسم می‌کند. متون زیر یک نمودار میخی برای داده‌های زیر تولید می‌کند.

$$x[0] = 2.4, x[1] = 3.8, x[2] = -1.8, x[3] = 2.1, x[4] = 1.9$$

$$x[5] = 0.8, x[6] = 0.0, x[7] = -0.8, x[8] = 1.5$$

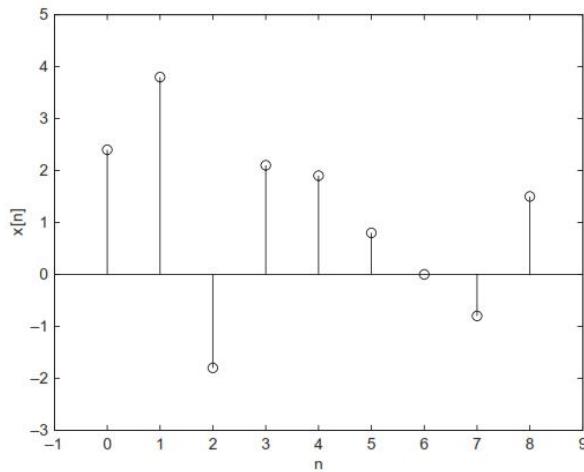
```
% Example_2_10.m
% This script uses the 'stem' command to plot discrete
% data.
clc; clear;
x = [2.4 3.8 -1.8 2.1 1.9 0.8 0.0 -0.8 1.5];
n = [ 0 1 2 3 4 5 6 7 8];
stem(n,x);
xlabel('n'), ylabel('x[n]');
axis([-1 9 -3 5]);
%-----
```

نمودار نتایج در شکل (۱۲.۲) نشان داده شده است.

مثال ۱۱.۲: تولید فایل داده برای استفاده در برنامه دیگر

```
% Example_2_11.m
```

^{۲۴}Stem Plot



شکل ۱۲.۲: نمودار میخی

```
% This program is used to create a data file that can be
% loaded in another program for use in a variety of ways.
clear; clc;
for n=1:20
    t(n)=n;
    y1(n)=n^2/10;
    y2(n)=n^3/100;
    y3(n)=n^4/1000;
end
fo=fopen('ydata.txt','w');
for i=1:20
    fprintf(fo,'%6.1f %10.5f %10.5f %10.5f \n',...
        t(i),y1(i),y2(i),y3(i));
end
% Running the program creates the data file 'ydata.txt':
fprintf('Data is saved to file ''ydata.txt''\n');
% Adding the clear statement removes all data from the
% work space:
clear;
%-----
```

مثال ۱۲.۲: بارگذاری داده تولید شده در برنامه دیگر

```
% Example_2_12.m
% This program demonstrates the use of the load command.
% A data file named ydata.txt (created in Example_2_11.m)
% is loaded into this program. Then, variables t,y1,y2,y3
% are extracted from the loaded data and plotted all on
```

```
% one graph.
clear; clc;
fprintf('Reading data from ''ydata.txt''\n');
load ydata.txt
t = ydata(:,1);
y1 = ydata(:,2);
y2 = ydata(:,3);
y3 = ydata(:,4);
plot(t,y1,t,y2,'--',t,y3,'-.');
xlabel('t'), ylabel('y1,y2,y3'),
title('y1, y2 and y3 vs. t'), grid;
legend('y1','y2','y3');
%-----
```

یک راه دیگر برای فرمان بار کردن `load` فرمان `dlmread` است. این فرمان یک فایل ASCII محدود را می‌خواند. تمام داده‌ها در فایل باید عددی باشند. در مثال ۱۲.۲، می‌توانستیم خطوطی که با `load ydata.txt` شروع و با `y3=ydata(:,4)` ختم می‌شوند را با

```
Y = dlmread('ydata.txt');
t = Y(:,1);
y1 = Y(:,2);
y2 = Y(:,3);
y3 = Y(:,4);
```

تعویض نمائیم.

مثال ۱۳.۲: نمودار نیمه‌لگاریتمی

```
% Example_2_13.m
% This program is a demonstration of a semilog plot
% command
clc; clear;
x=1:0.5:5;
for i=1:length(x)
    y(i)=3.0*exp(x(i));
end
semilogy(x,y), title('Semilog plot of y vs. x');
xlabel('x'), ylabel('y'),grid;
%-----
```

مثال ۱۴.۲: نمودار توابع مثلثاتی

```
% Example_2_14.m
% This script calculates functions sin(2x/3) and cos(3x+pi)
% for -pi <= x <= pi. The x domain is subdivided into 50
```

```
% subdivisions. The functions at each data point is
% calculated and stored in the vectors 'fsin' and 'fcos'.
% Then the two vectors are printed as a table to a file
% named 'output.txt'. Plots of both functions on the same
% axis are created. Finally, the maximum values of
% 'fsin' and 'fcos' are determined and printed.
clear; clc;
xmin=-pi; dx= 2*pi/50;
for i=1:51
    x(i)=xmin+(i-1)*dx;
    arg1=2*x(i)/3;
    arg2=3*x(i)+pi;
    fsin(i)=sin(arg1);
    fcos(i)=cos(arg2);
end
fo=fopen('output.txt','w');
% Table headings
fprintf(fo, ' x fsin fcos \n');
fprintf(fo, '-----\n');
for i=1:51
    fprintf(fo, '%10.5f %10.5f %10.5f \n', ...
        x(i),fsin(i),fcos(i));
end
fprintf(fo, '-----\n\n');
% Print maximum values:
fsin_max=max(abs(fsin));
fcos_max=max(abs(fcos));
fprintf(fo,'fsin_max=%10.5f\n',fsin_max);
fprintf(fo,'fcos_max=%10.5f\n',fcos_max);
% Plot data:
plot(x,fsin,x,fcos,'--');
xlabel('x'), ylabel('fsin,fcos'), grid,
title('fsin and fcos vs. x');
legend('fsin','fcos');
%-----
```

۱۰.۲ کار با ماتریس‌ها

مثال‌های اضافی زیر، استفاده از موارد زیر را نشان میدهد.

۱. حلقه‌های for تو در تو برای پر کردن ماتریس 3×3 از صفحه کلید.

۲. فرمان .input

۳. فرمان .load

۴. مثالی از نوشتن یک تابع.

.fscanf فرمان ۵.

مثال ۱۵.۲: ورودی داده به یک ماتریس با صفحه کلید

```
% Example_2_15.m
% Input a 3 by 3 matrix from the keyboard.
% Nested loops are used to create a 3 by 3 matrix.
% An element in the matrix is established by the indices
% of both the inner and outer loops.
clear; clc;
for n=1:3
    for m=1:3
        fprintf('n=%i m=%i ',n,m);
        a(n,m)=input('Enter a(n,m): ');
        fprintf('\n');
    end
end
% Print the resulting matrix:
a
%-----
```

مثال ۱۶.۲: پر کردن یک ماتریس از یک فایل

```
% Example_2_16.m
% Matrices A and B are loaded from a file named
% data216.txt, then printed to the screen. Matrices A and
% B are combined into matrix C, which is sent to function
% signal_out. This function subdivides matrix C back into
% matrices A and B and prints them out to the screen.
% Note: Before running this program, the data file
% data216.txt and function file signal_out.m need to be
% created (see below).
clear; clc;
n = 5;
load data216.txt
A = data216(:,1:n);
B = data216(:,n+1);
fprintf('Matrix A elements: \n');
A
fprintf('\n');
fprintf('Matrix B elements: \n');
B
C=[A B];
signal_out(C);
%-----
```

```
% Data file 'data216.txt'(do not include this header line)
8.77 2.40 5.66 1.55 1.0 -32.04
4.93 1.21 4.48 1.10 1.0 -20.07
3.53 1.46 2.92 1.21 1.0 -8.53
5.05 4.04 2.51 2.01 1.0 -6.30
3.54 1.04 3.47 1.02 1.0 -12.04
%-----
% signal_out.m
% This function accepts matrix A & matrix B and prints
% them to the screen and is used with Example 2.16.
function [] = signal_out(C)
n=5;
A=C(:,1:n);
fprintf('This output is from signal_out: \n\n');
fprintf('Matrix A elements: \n')
for i=1:n
    for j=1:n
        fprintf(' %8.3f ',A(i,j));
    end
    fprintf('\n');
end
fprintf('\n\n Matrix B elements: \n');
B=C(:,n+1);
for i=1:n
    fprintf(' %8.3f \n',B(i));
end
%-----
```

مثال ۱۷.۲: پر کردن یک ماتریس توسط fscanf

```
% Example_2_17.m
% This program uses the fscanf command to enter an M by N
% matrix into MATLAB. The M by N matrix is entered in
% column order. Thus, rows become columns and columns
% become rows. The script creates matrices A and B and
% then prints them out to the screen.
% Note: Before running this program, the data file
% data217.txt needs to be created (see data below).
clear; clc;
n=5;
A=zeros(n); B=zeros(5,1);
fin=fopen('data217.txt','r');
[A]=fscanf(fin,'%f',[n,n]);
fprintf(' Matrix A elements: \n');
A
[B]=fscanf(fin,'%f',[n]);
```

```

fclose(fin);
fprintf('\n');
fprintf(' Matrix B elements: \n');
B
fprintf('\n');
fprintf(' Matrix C elements: \n');
C=[A' B]
% Comparing the data217 data file with matrices A and B,
% we see that the data217 data file has been transposed
% (rows become columns and columns become rows).
%-----
% Data file 'data217'(do not include this header line)
8.77 2.40 5.66 1.55 1.0
4.93 1.21 4.48 1.10 1.0
3.53 1.46 2.92 1.21 1.0
5.05 4.04 2.51 2.01 1.0
3.54 1.04 3.47 1.02 1.0
-32.04 -20.07 -8.53 -6.30 -12.04
%-----

```

۱۱.۲ کار با تابع یک بردار

متلب اجازه میدهد تابع آرگومانش برداری باشد. مقدار برگشت می‌تواند نیز بردار باشد. برای مثال، (این برنامه ساده را در پنجره متن نوشته آنرا بنام vector-function.m ذخیره کرده و آنرا اجرا کنید):

```

% vector_function.m
clear; clc;
% Define vector x;
x = 0:30:360;
% Let y1 be the sine of a vector x where x is in degrees.
% Running sind on a vector will return a vector:
y1 = sind(x);
% Let y2(n) be the sine of the nth element of x. We will
% use a for loop to calculate each value y2(n) and then
% compare y1 and y2.
for n = 1:length(x)
    y2(n) = sind(x(n));
end
% Table headings
fprintf('x y1 y2 \n');
fprintf('-----\n');
for n = 1:length(x)
    fprintf('%5.1f %8.5f %8.5f \n',x(n),y1(n),y2(n))
end
%-----

```

آیا در خروجی تولید شده $y_1=y_2$ است؟
 می‌بینیم که در برخی متون، میتوانیم استفاده از حلقه `for` را با یکتابع برداری، که یک بردار تولید می‌کند، عوض کرده، بنابراین تعداد خطوط در متن را کاهش دهیم. این مفهوم در متن `vector_function.m` نمایش داده شده است. البته، اگر متن عملیات ریاضی دوتابع (مانند ضرب دوتابع برداری) لازم داشته باشد، در آن صورت عملیات عملکرد یک عنصر در یک عنصر نیاز خواهد داشت. عملیات عنصر در عنصر در فصل سوم بحث خواهد شد.

۱۲.۲ مثال‌های اضافی از کاراکترها و رشته‌ها

مثال‌های اضافی زیر نشانده‌ند

- چاپ یک رشته کاراکتر است.
- استفاده عبارت سویچ `switch` است.
- استفاده از نرده‌بان `if-elseif` است.
- استفاده از حلقه `for` برای گزینش محدوده مناسب در تشکیل درجه ماتریس یا درون‌یابی^{۲۵} است.

مثال ۱۸.۲: چاپ یک رشته کاراکتر از یک حلقه

```
% Example_2_18.m
% Sometimes you might wish to print out a string of
% characters in a loop. This can be done by declaring a
% 2-D character string matrix as shown in this example.
% Note that all row character strings must have the same
% number of columns and that character strings must be
% enclosed by single quotation marks.
clear; clc;
parts=['Internal modem '
       'Graphics adapter'
       'CD drive '
       'DVD drive '
       'Floppy drive '
       'Hard disk drive '];
for i=1:5
    fprintf('%16s \n', parts(i,1:16));
end
%-----
```

مثال ۱۹.۲: چاپ رشته‌ای در صفحه نمایش یا در یک فایل بر اساس استفاده از ورودی

```
% Example_2_19.m
% This example is a modification of Example 2.18. The
```

^{۲۵}interpolation

```
% program asks the user if they want to have the output
% go to the screen or to a file. This example illustrates
% the use of the switch statement.
clear; clc;
char=['Internal modem           '
      'External modem          '
      'Graphics circuit board'
      'CD drive                '
      'Hard disk drive        '];
fprintf('Choose whether to send the output to the\n');
fprintf('screen or to a file named ''output.txt''.\n');
fprintf('\n');
fprintf('Enter ''s'' for screen or ''f'' for file\n');
var = input('(without quotes): ','s');
switch(var)
    case 's'
        for i=1:5
            fprintf('%22s \n',char(i,1:22));
        end
    case 'f'
        fo=fopen('output.txt','w');
        for i=1:5
            fprintf(fo, '%22s \n',char(i,1:22));
        end
        fclose(fo);
    otherwise
        fprintf('You did not enter an ''s'' or an ');
        fprintf('''f'', try again.\n');
    end
%-----
```

مثال ۲۰.۲: نردنی if-elseif

```
% Example_2_20.m
% This example uses the if-elseif ladder.
% The program determines a letter grade depending on the
% score the user enters from the keyboard.
clear; clc;
gradearray=['A'; 'B'; 'C';'D';'F'];
score=input('Enter your test score: ');
fprintf('Score is: %i \n',score);
if score > 100
    fprintf('Error: score is out of range.\n');
    fprintf('Rerun program.\n');
    break;
```

```

elseif (score >= 90 && score <= 100)
    grade=gradearray(1);
elseif (score >= 80 && score < 90)
    grade=gradearray(2);
elseif (score >= 70 && score < 80)
    grade=gradearray(3);
elseif (score >= 60 && score < 70)
    grade=gradearray(4);
elseif (score < 60)
    grade=gradearray(5);
end
fprintf('Grade is: %c \n', grade);
%-----

```

مثال ۲۱.۲: استفاده از ورودی در تعیین رتبه

```

% Example_2_21.m
% This program determines a letter grade depending on the
% score the user enters from the keyboard. This version
% uses a loop to determine the correct interval of
% interest. For a large number of intervals, this method
% is more efficient (fewer statements) than the method in
% Example 2.20
clear; clc;
gradearray=['A'; 'B'; 'C'; 'D'; 'F'];
sarray=[100 90 80 70 60 0];
score=input('Enter your test score: ');
fprintf('Score is: %i \n', score);
% The following 2 statements are needed for the case
% when score = 100.
if score == 100
    grade=gradearray(1);
else
    for i=1:5
        if (score >= sarray(i+1) && score < sarray(i))
            grade=gradearray(i);
        end
    end
end
fprintf('Grade is: %c \n', grade);
%-----

```

مثال ۲۲.۲: چاپ آرایه‌ای کاراکتر

```
% Example_2_22.m
```

```
% This program determines the letter grades of several
% students. Student's names and their test scores are
% entered in the program. This example uses nested 'for'
% loops and an 'if' statement to determine the correct
% letter grade for each student.

clear; clc;
gradearray=['A'; 'B'; 'C'; 'D'; 'F'];
sarray=[100 90 80 70 60 0];
Lname=['Smith      '
       'Lambert    '
       'Kurtz      '
       'Jones      '
       'Hutchinson '
       'Diaz        '];
Fname=['Joe        '
       'Jane       '
       'Howard     '
       'Martin     '
       'Peter      '
       'Carlos     '];
score=[84; 72; 93; 64; 81; 75];
% The score = 100 is treated separately.

for j=1:6
    if score(j) == 100
        grade(j)=gradearray(1);
    else
        for i=1:5
            if score(j)>=sarray(i+1) && score(j)<sarray(i)
                grade(j)=gradearray(i);
            end
        end
    end
end
fprintf('Last name      First name      Grade \n');
fprintf('-----\n');
for j=1:6
    fprintf('%12s    %10s      %c \n', ...
            Lname(j,1:12), Fname(j,1:10), grade(j));
end
%-----
```

مثال ۲۳.۲: تابع خودنوشتاری و آرایه‌ای از کاراکتر

```
% Example_2_23.m
% This program uses func_grade.m (defined below) to
```

```
% determine the grade of a student. The input to the
% function is score and the function returns the grade
% to the calling program.
clear; clc;
Lname=['Smith      '
       'Lambert    '
       'Kurtz      '
       'Jones      '
       'Hutchinson'
       'Diaz       '];
Fname=['Joe       '
       'Jane      '
       'Howard    '
       'Martin    '
       'Peter     '
       'Carlos    '];
% The scores of the students named above are given in the
% vector score in the order listed in Lname.
score = [84; 72; 93; 64; 81; 75];
fprintf('Last name    First name    Grade \n');
fprintf('-----\n');
for j=1:6
    grade(j)=func_grade(score(j));
    fprintf('%12s %10s %c \n', ...
            Lname(j,1:12), Fname(j,1:10), grade(j));
end
%-----
% func_grade.m
% This function works with Example_2_23.m
function grade=func_grade(score)
gradearray=['A'; 'B'; 'C'; 'D'; 'F'];
sarray=[100 90 80 70 60 0];
if score == 100
    grade=gradearray(1);
else
    for i=1:5
        if (score >= sarray(i+1) && score < sarray(i))
            grade=gradearray(i);
        end
    end
end
%-----
```

۱۳.۲ درون‌یابی و تابع interp1 متلب

برای خیلی از پدیده‌های فیزیکی، اغلب در می‌یابیم که خواص ماده‌ای را می‌توان بر حسب تابعی از سایر متغیرها محاسبه کرد. برای مثال، قابلیت تحرک^{۲۶} الکترون‌ها درون سیلیکن وابسته به دما بوده و اغلب بصورت اندازه‌گیری جداول آزمایشگاهی برای چند درجه حرارت تهیه می‌شود. اگر بخواهیم قابلیت تحرک را در دمائی که در جدول نیست تعیین کنیم، لازم است درون‌یابی^{۲۷} نمائیم. فرمول کلی درون‌یابی خطی چنین است:

$$\mu = \mu_1 + \frac{(T - T_1)(\mu_2 - \mu_1)}{T_2 - T_1}$$

که در آن T_2 و T_1 دو دمای اندازه‌گیری شده در اطراف T ، و μ_2 و μ_1 قابلیت تحرک بترتیب در دو دمای T_2 و T_1 است. مثال بعدی محاسبه قابلیت تحرک را برای دما دلخواه بر اساس مقادیر جدول معلوم در درجه‌حرارت‌های معلوم را نشان میدهد.

مثال ۲۴.۲: درون‌یابی با استفاده از یک تابع ناشناس

```
% Example_2_24.m
% This program uses an anonymous function to linearly
% interpolate the electron mobility of silicon. Measured
% values of the mobility (cm^2/V-s) vs temperature (K)
% are given [2].
clear; clc;
% Anonymous function (avoids creating an extra .m file):
Muf = @(T,T1,T2,Mu1,Mu2) (Mu1+(T-T1)*(Mu2-Mu1)/(T2-T1));
Tt = [100 150 200 250 300 350 400 450];
Mut = [19650 7427 3723 2180 1407 972.0 705.5 531.8 ];
fprintf('This program interpolates for the electron\n');
fprintf('mobility (Mu) at a specified temperature T.\n');
fprintf('The valid temperature range is 100-450K.\n\n');
T=input('Enter T at which Mu is to be determined: ');
% Determining the closest surrounding temperatures to
% temperature T.
for n=1:length(Tt)-1
    if T >= Tt(n) && T <= Tt(n+1)
        T1=Tt(n); T2=Tt(n+1); Mu1=Mut(n); Mu2=Mut(n+1);
    end
end
Mu=Muf(T,T1,T2,Mu1,Mu2);
fprintf(' T=%.1f (K) Mu=%.3f (cm^2/V-s) \n',T,Mu);
%-----
```

متلب تابعی دارد بنام interp1 که درون‌یابی را انجام میدهد. مثال بعدی استفاده از این تابع را برای درون‌یابی قابلیت تحرک نمایش میدهد. بعلاوه، تابع interp1 متلب را می‌تواند چندین بار درون‌یابی را فقط با یکبار خواندن تابع انجام دهد.

^{۲۶}mobility

^{۲۷}interpolate

دستور العمل برای interp1 چنین است:

$$Y_i = \text{interp1}(X, Y, X_i)$$

که در آن X و Y مجموعه از نقاط داده x, y ، و X_i مجموعه ای از مقادیر x است که مجموعه Y_i مقادیر y را بوسیله درون یابی خطی میبایست تعیین کند. آرایه X و Y باید دارای طول یکسانی باشند. تابع interp1 را می‌توان برای روش‌های درون یابی سایر درون یابی خطی بکار برد. این در فصل نهم در زمینه برآذش منحنی آورده خواهد شد.^{۲۵}

مثال ۲۵.۲: درون یابی با استفاده از interp1 و ذخیره کردن در فایل .mat

```
% Example_2_25.m
% This program uses MATLAB's function interp1 to
% interpolate for the electron mobility of silicon.
% Measured values of the mobility (cm^2/V-s) vs
% temperature (K) are given [2]. Table values for
% temperature are specified in matrix T_data. Table
% values for mobility are specified in Mu_data. The
% temperature at which the electron mobility is to be
% determined is entered from the keyboard.
clear; clc;
T_data = [ 100 150 200 250 300 350 400 450 ];
Mu_data = [ 19650 7427 3723 2180 1407 972.0 705.5 531.8];
fprintf('This program interpolates for the electron\n');
fprintf('mobility (Mu) at a specified temperature T.\n');
fprintf('The valid temperature range is 100-450K.\n\n');
T=input('Enter T at which Mu is to be determined: ');
Mu = interp1(T_data,Mu_data,T);
fprintf(' T=%.1f (K)    Mu=%.3f (cm^2/V-s)\n\n', T, Mu);
% The data T_data and Mu_data can be saved in a data file
% and loaded later in another program. By the use of the
% 'clear' command the data file is removed from the
% workspace. Another script can bring the data back into
% the workspace by the load command. The script can then
% use the data for some other purpose. We want T_data and
% Mu_data to be column vectors, so save the transpose.
T_Mu_data = [T_data' Mu_data'];
save T_Mu_data;
% T_Mu_data is saved in a .mat file and will be used
% in Example 2.25.
fprintf('Mobility vs. temperature data saved to file ');
fprintf('T_Mu_data.mat\n');
clear;
% T_Mu_data has been removed from the work space
% T_Mu_data will be used in Example_2_26.
%-----
```

^{۲۵}Curve fitting

اگر بخواهیم قابلیت تحرک الکترون‌ها را در چندین دمای مختلف داشته باشیم، می‌توانیم مثال ۲۵.۲ را با تعویض تمام خطوط بعداز (و شامل) خطوط

```
fprintf('This program interpolates for the electron \n');
fprintf('mobility (mu) at a specified temprature T. \n');
```

با خطوط زیر:

```
T2 = [432 182 343 210 385];
fprintf('This program interpolates for the electron \n')
fprintf('mobility (Mu) at a specified temprature T. \n');
fprintf('specified temperatures,T2. \n');
fprintf('The allowable temperature range is 100-450K. \n\n');
Mu = interp1(T_data,Mu_data,T2);
fprintf('T2(K) Mu(cm^2/V-s) \n');
fprintf('-----\n');
for i = 1:5
    fprintf('%6.1f %9.3f \n',T2(i),Mu(i));
end
%-----
```

اصلاح کنیم.

مثال ۲۶.۲: درون یابی داده‌ها انبار شده در یک فایل .mat

```
% Example_2_26.m
% This script is a sample of a plot program loading data
% obtained in another program. Make sure you run Example
% 2.25 before running this example.
clear; clc;
fprintf('Loading data from file "T_Mu_data.mat"\n');
load T_Mu_data
T= T_Mu_data(:,1);
Mu= T_Mu_data(:,2);
plot(T,Mu);
xlabel('T'), ylabel('Mu'), title('Mu vs. T'), grid;
%-----
```

۱۴.۲ تابع textscan متلب

بعضی وقت‌ها می‌خواهیم فایل داده‌ای را در درون برنامه‌ای که شامل هم داده عددی و هم رشته‌ای کاراکتر است را بخوانیم. تابع textscan متلب بهترین گزینه در این وضع است. دستواعمل:

```
C = textscan(fid, format)
```

تابع داده را از یک باز کننده متن که با `fid` مشخص می‌شود آنرا در آرایه سلول `C` می‌خواند. فرمت آن بصورت `%f`, `%d`, `%c`, `%s` و غیره است. تعداد مشخص‌کننده فرمت تعداد سلول‌ها درون آرایه `C` را تعیین می‌کند. هر سلول شامل تعداد خطوطی است که شامل فایل داده بوده و نوع

آن با بیان فرمت مشخص است. مشخص کننده رشته کاراکتر نیز شامل `%q` بوده، که رشته کاراکتر با علامت نقل قول ”بسته است. برای یک کاراکتر تکی از `%c` استفاده کنید.

توجه: برای رجوع به محتوای یک سلول، تمام محتوای اعداد سلول با {} را در بر می‌گیرد. به مثال زیر مراجعه نمایید.

اگر میخواهید N خط از فایل داده باز شده را بخوانید، از
`C = textscan(fid, format, N)`

استفاده کنید.

مثال ۲۷.۲: بار کردن مخلوط داده‌های عددی و متن از یک فایل

```
% Example_2_27.m
% Load the product data from inv4.txt into the arrays
% 'sku', 'desc', and 'cost', and print.
% sku, desc and cost stands for item #,
% item description and item cost respectively.
clear; clc;
% Note: inventory227.txt is defined below.
fid=fopen('inventory227.txt');
C = textscan(fid, '%d %14c %f', 5);
% Contents of cell block C contains 1 row and 3 columns
sku = C{1};
desc = C{2};
cost = C{3};
fclose(fid);
count = length(sku);
for i=1:count
    fprintf('%5i %14s %6.2f\n\n', ...
        sku(i), desc(i,1:14), cost(i));
end
%-----
% inventory227.txt file (do not include this header line)
1 hammer      2.58
2 plier       1.20
3 screwdriver 1.56
4 soldering iron 3.70
5 wrench      2.60
%-----
```

مثال ۲۸.۲: مثالی دیگر از بار کردن مخلوطی از متن و عدد از یک فایل

```
% Example_2_28.m
% Read product inventory into a matrix using textscan.
% The file 'inventory228.txt' is listed below.
clc; clear;
fid=fopen('inventory228.txt');
C = textscan(fid,'%d %q %f' );
```

```
% Need to designate contents in cell C{2} as characters.
sku = C{1};
desc = char(C{2});
cost = C{3};
count = length(sku);
fprintf(' sku# description cost \n');
fprintf('-----\n');
for i=1:count
    fprintf('%4i %14s %6.2f \n\n', ...
        sku(i), desc(i,1:14), cost(i));
end
fclose(fid);
%-----
% inventory228.txt file (Do not includ this header line)
1 "hammer" 2.58
2 "plier" 1.20
3 "screwdriver" 1.56
4 "soldering iron" 3.70
5 "wrench" 2.60
%-----
```

۱۵.۲ انتقال داده متلب به اکسل

متلب می‌تواند داده‌هایش را مستقیماً در یک نرمافزار صفحه‌گسترده^{۲۹} از قبیل اکسل Excel بنویسد. برای انجام این امر، مراحل زیر را دنبال کنید:

۱. در برنامه متلب خود، ماتریسی با داده خروجی، مثلاً M تولید کنید.

۲. داده خود را در یک فایل اکسل با عبارت زیر وارد کنید:

```
xlswrite('filename.xls',M);
```

این باعث می‌شود که یک فایل اکسل بنام filename.xls در داده، M ، شما باز شود که شما بعداً می‌توانید درون اکسل آنرا ویرایش کنید. (به شکل ۱۳.۲ مراجعه کنید).

۳. اکنون میتوانید تمام فرمتهای فرامین کار با داده که در اکسل است استفاده کنید. برای مثال، عنوان جدول اضافه کنید، فایل اکسل را باز کنید، مکان نما را در بالای سمت چپ سلول (A1) قرار داده، و عنوان جدول را در هر ستون سلولها تایپ کنید. برای اضافه کردن ستون دوم، مکان نما را در سلول B13 قرار داده و $=SUM(B3:B13)$ را تایپ کنید. ضمناً میتوانید کلمات درشت، کج، و سایر حاشیه‌ها را اعمال کنید.

۱۶.۲ اشکال‌زدایی یک برنامه

معمول است وقتی برنامه‌ای نوشته می‌شود، خطاهای چاپی از قبیل حذف پرانتر، فراموش کردن ویرکول در آرایه دو بعدی، و غیر پیش می‌آید. این نوع اشکالات را خطای دستوری^{۳۰} می‌نامند. وقتی این

^{۲۹}spreadsheet

^{۳۰}syntax error

| A | B | C |
|----|-----|------|
| 1 | 0.1 | 0.01 |
| 2 | 0.4 | 0.08 |
| 3 | 0.9 | 0.27 |
| 4 | 1.6 | 0.64 |
| 5 | 2.5 | 1.25 |
| 6 | 3.6 | 2.16 |
| 7 | 4.9 | 3.43 |
| 8 | 6.4 | 5.12 |
| 9 | 8.1 | 7.29 |
| 10 | 10 | 10 |
| 11 | | |
| 12 | | |

شکل ۱۳.۲: مثالی از ارسال خروجی به اکسل.

اتفاق می‌افتد، متلب یک پیام خطای روی خطی که خط رخ داده نشان میدهد. البته، حالت‌های وجود دارد که این گونه خطای دستوری وجود ندارد، اما برنامه اجرا نمی‌شود یا جواب غلط واضح میدهد. وقتی این اتفاق می‌افتد، شما می‌توانید از اشکال زدایی^{۴۱} متلب استفاده کنید. این صورت اشکال زدایی بشما اجازه میدهد تا نقطه قطع در برنامه خود بوجود آورید. برنامه تا آن خط که نقطه قطع برنامه است اجرا می‌شود. برای تعیین نقطه قطع برنامه، روی ستون باریک نزدیک شماره خط مورد نظر خود کلیک کنید. یک دایره کوچک قرمز رنگ در نزدیکی خط نقطه قطع، مانند شکل (۱۴.۱)، ظاهر می‌شود.

اکنون برنامه تا این خط (نه خود خط)، مانند شکل (۱۵.۲) (در نقطه فرمان چاپ ماتریس *A*) اجرا می‌شود. علامت $>> K$ نمایشگر این است که شما در وضع اشکال زدایی هستید. سپس می‌توانید روی فهرست منو جعبه‌ابزار روی debug کلیک کرده و از چندین گزینه در روی منو پائین آورده شده (به شکل ۱۶.۲ مراجعه کنید) انتخاب کنید. اگر برنامه شاملتابع خودنوشته باشد، می‌توانید در روی خط زمان تابع، محاسبه را ادامه دهید. برای این کار، ابتداء لازم است تمام نقاط قطع برنامه را در برنامه اصلی پاک کنید، سپس گزینه Step In را در منو debug (به شکل ۱۶.۲ مراجعه کنید) انتخاب نمائید. گزینه Step Out دوباره به برنامه اصلی برミگردد. با انتخاب Clear Break points in All Files تمام نقاط قطع از برنامه حذف می‌شود.

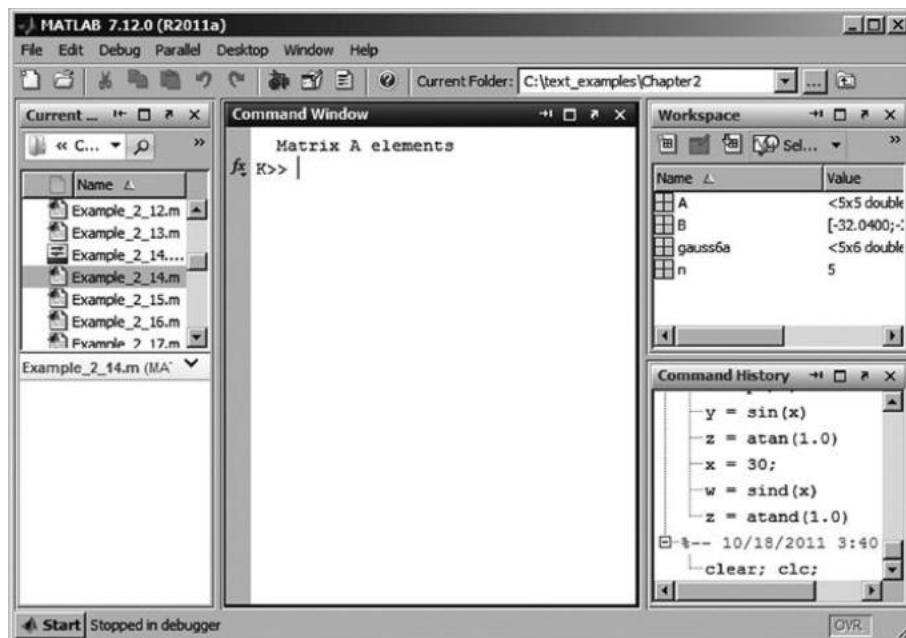
^{۴۱}Debugging

```

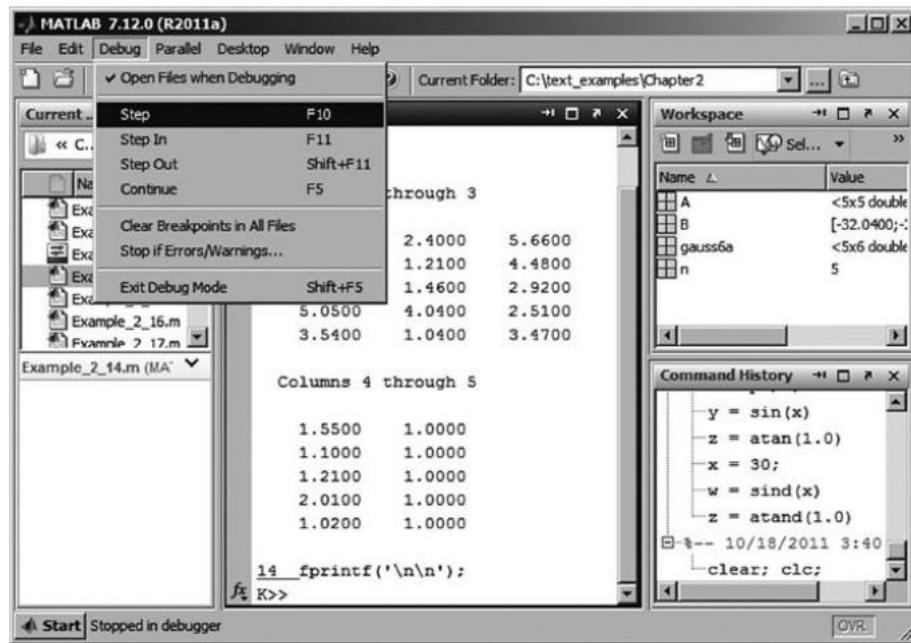
Editor - C:\text_examples\Chapter2\Example_2_14.m
File Edit Text Go Cell Tools Debug Desktop Window Help
... - [1.0] + [1.1] * [%] [%] [ ]
1 % Example_2_14.m
2 % Matrices A and B are loaded from a file named gauss6a.txt, then printed
3 % to the screen. Matrices A and B are combined in matrix C, which is sent
4 % to function moutput. This function subdivides matrix C into matrices A
5 % and B and prints them out to the screen.
6 clear; clc;
7 n=5;
8 load gauss6a.txt
9 A=gauss6a(:,1:n);
10 B=gauss6a(:,n+1);
11 fprintf(' Matrix A elements \n');
12
13 A
14 fprintf('\n\n');
15 fprintf(' Matrix B elements \n');
16
17 B
18 C=[A B];
19 moutput(C);
20

```

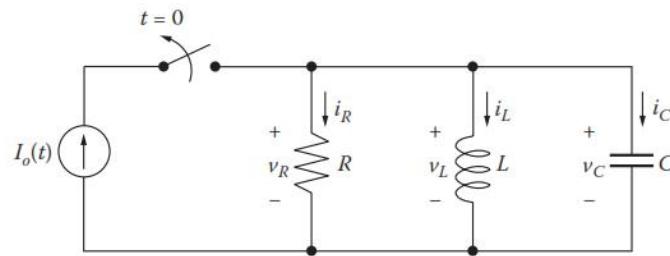
شكل ١٤.٢: تعیین نقطه قطع برنامه



شكل ١٥.٢: اجرای برنامه تا نقطه قطع و سپس توقف.



شکل ۱۶.۲: منو اشکال زدایی



شکل ۱۷.۲: مدار RLC موازی

۱۷.۲ مدار RLC موازی

مدار مقاومت، سیمپیج، خازن (RLC) مسئله بنیادین در مهندسی برق است که ما بارها آنرا در این کتاب ملاقات می‌کنیم (به شکل ۱۷.۲ و پیوست آ مراجعة کنید). این بخش معادلات حاکم که توصیف کننده ولتاژ و جریان در عناصر مدار است را بدست می‌آورد. مدار با منبع جریان $I_0(t)$ از طریق کلید که در لحظه $t = 0$ باز می‌شود، تغذیه می‌گردد. در حال حاضر در باره تعريف $I_0(t)$ بغيير از آنکه شرایط اولیه را برای عناصر تدارک می‌نماید، نظری نداریم. ما تحلیل را با تشکیل رابطه ولتاژ-جریان برای مقاومت، سیمپیج، و خازن شروع می‌کنیم:

$$v_R = Ri_R \quad \text{مقاآمت: (قانون اهم)} \quad (1.2)$$

$$v_L = L \frac{di_L}{dt} \quad \text{سیمپیج:} \quad (2.2)$$

$$i_C = C \frac{dv_C}{dt} \quad \text{خازن :} \quad (3.2)$$

که در آن R , L و C مقادیر عناصر بترتیب برای مقاومت (بر حسب اهم)، سیم پیچ (بر حسب هانری)، و خازن (بر حسب فاراد) و ولتاژ v و جریان i در شکل (۱۷.۲) تعریف شده است. بعلاوه، فرض بر این است که شرایط اولیه عناصر L و C یعنی $i_{L(0)}$ و $v_{C(0)}$ در لحظه‌ای که کلید باز می‌شود معلوم است. قانون جریان کیرشهف (KCL) می‌گوید که جمع جریان‌ها در هر گره مداری صفر است. با اعمال KCL در گره بالای سمت راست مدار داده شده

$$i_R + i_L + i_C = 0 \quad (4.2)$$

همچنین، بعلت موازی بودن مدار خواهیم داشت

$$v_R = V_L = v_C = 0 \equiv v \quad (5.2)$$

با قرار دادن رابطه (۵.۲) در رابطه (۲.۲) بدست می‌آید:

$$\frac{di_L}{dt} = \frac{1}{L}v \quad (6.2)$$

و قرار دادن رابطه (۱.۲) و (۳.۲) در معادله (۴.۲) داریم:

$$\frac{dv}{dt} = -\frac{1}{RC}v - \frac{1}{C}i_L \quad (7.2)$$

رابطه (۶.۲) و (۷.۲) دستگاهی متشکل از دو معادله دیفرانسیل مرتبه اول با دو مجهول است. می‌توانیم این‌ها را با هم ترکیب کرده و با مشتق‌گیری از دو طرف رابطه (۷.۲)، یک معادله دیفرانسیل مرتبه دوم بدست اوریم:

$$\frac{d^2v}{dt^2} = -\frac{1}{RC} \frac{dv}{dt} - \frac{1}{C} \frac{di_L}{dt} \quad (8.2)$$

با قرار دادن رابطه (۸.۲) در رابطه (۸.۲) معادله دیفرانسیل مرتبه دوم زیر را خواهیم داشت:

$$\frac{d^2v}{dt^2} + \frac{1}{RC} \frac{dv}{dt} + \frac{1}{LC}v = 0 \quad (9.2)$$

برای حل این معادله دیفرانسیل مرتبه دوم همگن، ما بدنیال تابعی برای $v(t)$ می‌گردیم که مشتقات $v'(t)$ و $v''(t)$ دوباره خود تابع $v(t)$ را تولید کنند. تابعی که این شرایط را برآورده می‌کند $Ae^{\alpha t}$ است که در آن A و α دو مقدار ثابت دلخواه هستند. بنابراین اگر فرض کنیم $v = Ae^{\alpha t}$ در این صورت: $v'(t) = \alpha Ae^{\alpha t}$ و $v''(t) = \alpha^2 Ae^{\alpha t}$

با قرار دادن این جملات در معادله (۹.۲) بدست می‌آید

$$(\alpha^2 + \frac{1}{RC}\alpha + \frac{1}{LC})Ae^{\alpha t} = 0 \quad (10.2)$$

چون $e^{\alpha t} \neq 0$ است، باید معادله درجه دوم زیر را حل کنیم

$$\alpha^2 + \frac{1}{RC}\alpha + \frac{1}{LC} = 0 \quad (11.2)$$

که جواب‌های آن عبارتند از

$$\alpha = -\frac{1}{2RC} \pm \sqrt{(\frac{1}{2RC})^2 - \frac{1}{LC}} \quad (12.2)$$

بنابراین، دو جواب، معادله دیفرانسیل را برای دو مقدار α رابطه (۱۲.۲) راضی می‌کند. جواب کلی معادله (۹.۲) جمع دو جواب است:

$$v = A \exp \left[-\frac{1}{2RC}t + \sqrt{(\frac{1}{2RC})^2 - \frac{1}{LC}}t \right] + B \exp \left[-\frac{1}{2RC}t - \sqrt{(\frac{1}{2RC})^2 - \frac{1}{LC}}t \right]$$

یا

$$v = \exp\left(-\frac{1}{2RC}t\right) \left[A \exp\left(+\sqrt{\left(\frac{1}{2RC}\right)^2 - \frac{1}{LC}}t\right) + B \exp\left(-\sqrt{\left(\frac{1}{2RC}\right)^2 - \frac{1}{LC}}t\right) \right] \quad (13.2)$$

که در آن $A = \exp(x) = e^x$ و $B = \text{ثابت هستند که بستگی به شرایط اولیه مدار دارند.}$

• الف- فرامیرا^{۳۲}: اگر $\frac{1}{LC} > \frac{1}{2RC}$ باشد، در این صورت جواب بصورت نمائی میرا می‌شود.

• ب- زیرمیرا^{۳۳}: اگر $\frac{1}{LC} < \frac{1}{2RC}$ باشد، در این صورت جواب بصورت نوسانی (سینوسی) میرا می‌شود.

برای حالت زیرمیرا، می‌توانیم رفتار سینوزوئیدال را با اعمال اتحاد $e^{jx} = \cos x + j \sin x$ و اتحاد $e^{-jx} = \cos x - j \sin x$ در رابطه (13.2) نشان دهیم.

$$v = \exp\left(-\frac{1}{2RC}t\right) \left[A' \cos\left(\sqrt{\frac{1}{LC} - \left(\frac{1}{2RC}\right)^2} t\right) + B' \sin\left(\sqrt{\frac{1}{LC} - \left(\frac{1}{2RC}\right)^2} t\right) \right] \quad (14.2)$$

که در آن ضرایب A' و B' از شرایط اولیه بدست می‌آیند.

اگر $\frac{1}{LC} = \frac{1}{2RC}$ باشد، در این صورت زیر جذر صفر شده، و سیستم را میرائی بحرانی می‌نامند. برای این حالت^{۳۴}، جواب برابر است با

$$v = (A + Bt) \exp\left(-\frac{1}{2RC}t\right) \quad (15.2)$$

معادلاتی که در این بخش تدوین شد در پژوهه‌هایی که از ۷.۲ شروع می‌شود، کاربرد دارد.

مسائل

۱.۲ • ثابت کنید که بسط سری تیلور $\cos x$ حول نقطه $x = 0$ برابر است با:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

۲.۲ • ثابت کنید که بسط سری تیلور $\sin x$ حول نقطه $x = 0$ برابر است با:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

۳.۲ • با استفاده از بسط سری تیلور e^x حول نقطه $x = 0$ (به مثال ۱.۲ مراجعه کنید)، ثابت کنید که $e^{jx} = \cos x + j \sin x$ است. در اینجا $j = \sqrt{-1}$.

پژوهه‌ها

^{۳۲}Overdamped

^{۳۳}Underdamped

^{۳۴}critically damped

۱.۲ پروژه

با استفاده از بسط سری تیلور $\cos x$ حول نقطه $x = 0$ (به مسئله ۱.۲ مراجعه کنید)، برنامه‌ای کامپیوتری در متلب بنویسید که مقدار $\cos x$ را از $-\pi \leq x \leq \pi$ در قدم‌های 0.1π محاسبه نماید. سری را با تعیین هر جمله بر طبق معادله زیر عامل ضرب \times جمله $(k+1) = \text{جمله}(k)$

در اینجا لازم است عامل ضرب را تعیین کنید. تا حدود پنجاه جمله را بکار برد؛ البته، وقتی آخرین جمله فقط شش‌مین رقم اعشار را در جواب تاثیر میگذارد قبول کنید. ضمناً مقدار $\cos x$ را با استفاده تابع متلب $\cos()$ در همان محدوده محاسبه کنید. جدول نتایج خود را در فایلی مانند جدول (۲.۲) چاپ کنید. برای $\cos x$ شش رقم اعشار بکار برد.
هم‌چنان، نمودار $\cos x$ را در محدوده $-\pi \leq x \leq \pi$ رسم کنید.

جدول ۲.۲: فرمت جدول پروژه ۱.۲

| x | $\cos x$ با تابع $\cos()$ | $\cos x$ با روش سری | $\cos x$ با جملات در سری |
|-----------|------------------------------|------------------------|-----------------------------|
| -1.0π | | | |
| -0.9π | | | |
| -0.8π | | | |
| \vdots | | | |
| 0.9π | | | |
| 1.0π | | | |

۲.۲ پروژه

با استفاده از بسط سری تیلور $\sin x$ حول نقطه $x = 0$ (مسئله ۲.۲) برنامه‌ای کامپیوتری بزیان متلب بنویسید که مقدار $\sin x$ در محدوده $-\pi \leq x \leq \pi$ در قدم‌های 0.1π محاسبه نماید. سری را با استفاده تابع factorial متلب f و با استفاده از پنجاه جمله محاسبه کنید. البته، پس از این که آخرین جمله بیش از شش رقم اعشار روی جواب تاثیر میگذارد آنرا قطع کنید. هم‌چنان مقدار $\sin x$ را با استفاده از تابع $\sin()$ متلب در همان محدوده و فواصل محاسبه کنید. جدولی مانند پروژه ۱.۲ تولید کرده و بجای $\cos x$ تابع $\sin x$ را قرار دهید. ضمناً نمودار $\sin x$ را در محدوده $-\pi \leq x \leq \pi$ رسم کنید.

۳.۲ پروژه

برنامه‌ای کامپیوتری بزیان متلب نوشته که تابع زیر را در محدوده $0.9 \leq x \leq 0.1$ در قدم‌های ۰.۱ محاسبه نماید:

۱. از عبارت ریاضی.
۲. از بسط سری تا حداقل پنجاه جمله. البته، وقتی آخرین جمله بیش از شش رقم اعشار در جواب موثر است، آنرا قطع کنید.

$$f(x) = (1 + x^2)^{-1/2} = 1 - \frac{1}{2}x^2 + \frac{1 \cdot 3}{2 \cdot 4}x^4 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^6 + \frac{1 \cdot 3 \cdot 5 \cdot 7}{2 \cdot 4 \cdot 6 \cdot 8}x^8 - \dots$$

جدولی مانند جدول (۳.۲) با شش رقم اعشار برای $f(x)$ (در یک فایل) چاپ کنید.

جدول ۳.۲: فرمت جدول پروژه

| x | $f(x)$ با تابع ریاضی | $f(x)$ با روش سری | تعداد جملات در سری |
|------|-------------------------|----------------------|-----------------------|
| -0.9 | | | |
| -0.8 | | | |
| -0.7 | | | |
| ⋮ | | | |
| 0.7 | | | |
| 0.8 | | | |
| 0.9 | | | |

۴.۲ پروژه

برنامه‌ای کامپیوتری بزبان متلب نوشته که تابع

$$f(x) = (1+x)^{1/3}$$

را در محدوده $-0.9 \leq x \leq 0.9$ در قدمهای 0.1 محاسبه نماید:

۱. از عبارت ریاضی.

۲. از بسط سری تا حداکثر پنجاه جمله. البته، وقتی آخرین جمله بیش از شش رقم اعشار در جواب موثر است، آنرا قطع کنید.

از بسط سری تیلور استفاده کنید. رابطه بین جمله‌های سری چنین است

$$(k+1) \times \frac{1}{3} - (k-1) \times \frac{x}{k} = \text{جمله}(k)$$

برای $k = 1, 2, 3, \dots$

جدولی مانند جدول (۳.۲) با شش رقم اعشار برای $f(x)$ (در یک فایل) چاپ کنید.

۵.۲ پروژه

بسط بینم^{۳۴} $(1+x)^n$ که در آن n عدد صحیح است، بصورت زیر است:

$$(1+x)^n = 1 + nx + \frac{n(n-1)x^2}{2!} + \frac{n(n-1)(n-2)x^3}{3!} + \dots + \frac{n(n-1)(n-2)\dots(n-r+2)x^{r-1}}{(r-1)!} + \dots + x^n$$

برنامه متلبی بنویسید که $(1+x)^n$ را با هر دو روش سری و عبارت ریاضی برای $n = 10$ بازه $1.0 \leq x \leq 10.0$ بقدمهای 0.5 محاسبه کند. نتایج را در جدولی مطابق جدول (۴.۲) چاپ کنید.

۶.۲ پروژه

این پروژه شکل موج سینوزوئیدال $V_{peak-to-peak} = 340$ در فرکانس $60Hz$ بدون مولفه (بمرکزیت صفر ولت) را امتحان می‌کند.

۱. نمودار این شکل موج را در متلب با استفاده از تابع \sin در بازه $0 \leq t \leq 50$ میلی ثانیه در قدمهای 0.2 میلی ثانیه رسم کنید.

^{۳۴}binomial expansion

جدول ۵.۲: فرمت جدول پروژه ۴.۲

| x | $(1+x)^n$ با تابع ریاضی | $(1+x)^n$ با روش سری |
|------|----------------------------|-------------------------|
| 1.0 | | |
| 1.5 | | |
| 2.0 | | |
| 2.5 | | |
| : | | |
| 9.5 | | |
| 10.0 | | |

۲. مقدار مربع میانگین جذر (RMS) ولتاژ را برای این شکل موج با انجام عملیات زیر در متلب، محاسبه کرده و در صفحه نمایشگر چاپ کنید:

- الف: آنرا مربع کنید (بتوان دو کنید).

- ب: مقدار میانگین مربع شکل موج را توسط متوسطگیری تحت فاصله زمانی 100 میلی ثانیه محاسبه کنید.

- ج: از مقدار متوسط جذر بگیرید.

۳. در آمریکای شمالی، سرویس برق منازل معمولاً با دو ولتاژ سینوزوئیدال $340 V_{peak-to-peak}$ که هر کدار با دیگری 180° اختلاف فاز داردند تامین می شود (به شکل ۱۸.۲-الف رجوع شود) و به آن سرویس تک فاز میگویند. برای کاربردهای توان بالا (مثل، تهویه مطبوع، و اجاقها)، جریان از هر دو "پایه" اخذ می شود.

- الف: شکل موج اختلاف پتانسیل بین دو سیم سرویس تک فاز را رسم کنید.

- ب: مقدار RMS شکل موج اختلاف پتانسیل بین دو سیم را محاسبه کرده و آنرا روی صفحه نمایشگر نشان دهید.

۴. در خیلی از اماكن صنعتی سرویس برق سه فاز $340 V_{peak-to-peak}$ هر کدام با دیگری ۱۲۰° اختلاف فاز دارند، تامین می شود (به شکل ۱۸.۲-ب رجوع شود).

- الف: شکل موج اختلاف ولتاژ بین دو سیم سه فاز را رسم کنید.

- ب: مقدار RMS شکل موج اختلاف پتانسیل این شکل موج را محاسبه و در صفحه نمایشگر نشان دهید. اکثر دستگاهها مصرف کننده ولتاژ بالا این مقادیر ولتاژ RMS را که در بخش سوم یا چهارم محاسبه شده است را در روی دستگاه خود حک می کنند.

• پروژه ۷.۲

در یک مدار RLC موازی مقادیر عناصر بقرار زیر است

$$R = 100 \Omega, \quad L = 1 mH, \quad C = 1 \mu F$$

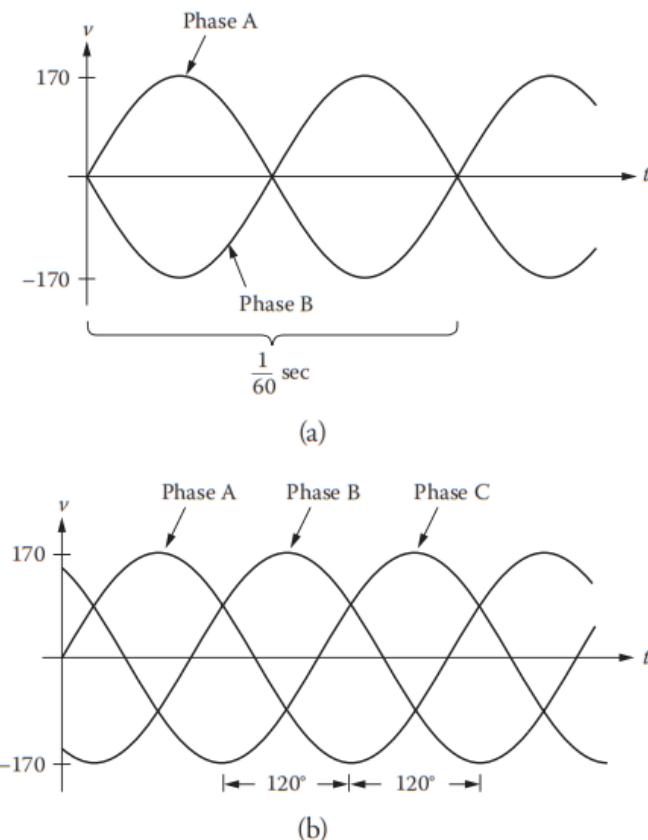
برای شرایط اولیه زیر، ضرایب A و B را تعیین کنید.

$$i_L(0) = 0.25 A, \quad v_C(0) = 6 V .1$$

راهنمایی: از طریق رابطه (۴.۲) $i_L(0)$ به $v(0)$ تبدیل کنید.

$$i_L(0) = -0.25 A, \quad v_C(0) = 6 V .2$$

$$i_L(0) = 0 A, \quad v_C(0) = 6 V .3$$



شکل ۱۸.۲: (الف) توان یک-فاز با دو سینوزوئیدال با اختلاف فاز 180° تحویل داده می‌شود. (ب) توان سه‌فاز بصورت سه سینوزوئیدال که هر کدام با دیگری 120° اختلاف فاز دارد، تامین می‌شود.

برای هر حالت، رابطه‌ای برای A و B بر حسب C, L, R ، $v_C(0)$ و $i_L(0)$ تعیین کنید و برنامه‌ای در متلب بنویسید که

- الف: ولتاژ $v(t)$ برای $0 \leq t \leq 500\mu sec$ بدست آورد.

- ب: جدولی تدوین کرده که v را بر حسب t در قدمهای $10\mu sec$ نشان دهد.

- ج: نمودار v را بر حسب t برای هر سه حالت در روی یک شکل رسم کنید. هر منحنی را با برچسب مقدار $i_L(0)$ مشخص نمایید.

۸.۲ پروژه

شکل موج ولتاژ مدار RLC که در بخش ۱۷.۲ تشریح شده بقرار زیر است.

$$v(t) = \pm Ae^{(-t/RC)} \quad (18.2)$$

اگر مقادیر عناصر بقرار زیر باشد؛

$$R = 5000 \Omega, \quad L = 1\mu H, \quad C = 10pF$$

$$i_L(0) = 0 A, \quad v_C(0) = 3.3 V$$

ولتاژ $v(t)$ را برای $0 \leq t \leq 50 nsec$ در قدمهای یک نانو ثانیه تعیین کنید. نمودار v بر حسب t را برای هر دوتابع نوسانی روی یک شکل رسم کنید.

• پروژه ۹.۲

بجای استفاده از v بعنوان متغیر حالت در تحلیل مدار RLC، میتوانیم i_L را با مشتق‌گیری از رابطه (۶.۲) و ترکیب با رابطه (۱۷.۲) بصورت معادله دیفرانسیل زیر بدست آوریم

$$\frac{d^2i_L}{dt^2} + \frac{1}{RC} \frac{di_L}{dt} + \frac{1}{LC} i_L = 0 \quad (17.2)$$

توجه کنید که معادله (۱۷.۲) همان صورت معادله (۱۲.۲) دارد و جواب این نیز به همان صورت است. برای حالت زیرمیرا،

$$i_L = \exp\left(-\frac{1}{2RC}t\right) \left[\alpha \cos\left(\sqrt{\frac{1}{LC} - \left(\frac{1}{2RC}\right)^2} t\right) + \beta \sin\left(\sqrt{\frac{1}{LC} - \left(\frac{1}{2RC}\right)^2} t\right) \right] \quad (18.2)$$

که در آن α و β از شرایط اولیه بدست می‌آید و دارای واحد آمپر است.

برای مقادیر عناصر $R=100\Omega$ ، $C=1\mu F$ ، $L=1mH$ و $v_C(0)=6V$ و $i_L(0)=0$ را

۱. ضرایب α و β را حل کنید.

۲. نمودار $i_L(t)$ برای بازه $0 \leq t \leq 500 \mu sec$ در قدمهای $0.5 \mu sec$ رسم کنید.

۳. نمودار $v_C(t)$ و $i_L(t)$ (در پروژه ۷.۲ حل شده است) روی یک محور رسم کنید. توجه کنید که وقتی v_C بیشینه است، i_L صفر و برعکس است. این نشان میدهد که چگونه انرژی بین خازن و سیم پیچ رد و بدل می‌شود.

• پروژه ۱۰.۲

صورت کلاسیک معادله دیفرانسیل درجه دوم بقرار زیر است:

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = 0 \quad (19.2)$$

که در آن ζ ضریب میرائی^{۳۶} و ω_n فرکانس طبیعی^{۳۷} است. میتوانیم حمل معادله (۱۷.۲) را با معادله (۱۹.۲) تطبیق داده و ضریب میرائي و فرکانس طبیعی را برای مدار RLC بدست آوریم. بنابراین،

$$\omega_n = \frac{1}{\sqrt{LC}} \quad \text{و} \quad \zeta = \frac{1}{2RC\omega_n} = \frac{\sqrt{LC}}{2RC} \quad (20.2)$$

توجه کنید که برای $\zeta < 1$ مدار زیرمیرا، و برای $\zeta > 1$ مدار فرامیرا و بازاء $1 = \zeta$ میرائي بحرانی است.

برای مقادیر عناصر $v_C(0)=6V$ ، $i_L(0)=0.25A$ و $C=1\mu F$ ، $L=1mH$ و

۱. مقدار مقاومت R_{crit} که مدار را به میرائي بحرانی مبدل تعیین کنید.

۲. جریان سیم پیچ i_L را نسبت به زمان برای سه مقدار $R=R_{crit}$ ، $R=5R_{crit}$ و $R=\frac{1}{2}R_{crit}$ رسم کنید. فواصل زمانی $500 \mu sec$ در قدمهای $0.5 \mu sec$ در نظر گیرید. هر سه شکل موج را روی یک نمودار رسم کنید.

• پروژه ۱۱.۲

می‌خواهیم مدار RLC که در بخش ۱۷.۲ توصیف شده را تحت جریان اعمالی ب نحوی که بجای باز کردن کلید در لحظه $t=0$ ، کلید را ببندیم. در این حالت، مجموع جریانها در گره سمت چپ بالا خواهد بود

$$i_R + i_L + i_C - I_0 = 0 \quad (21.2)$$

^{۳۶}damping factor

^{۳۷}natural frequency

توجه کنید که جمله I_0 بدین علت منفی است که در شکل بجای آن که خارج شود بسوی داخل جریان دارد (که این برای سه جریان دیگر درست است).

معادله حاکم خواهد بود

$$\frac{d^2 i_L}{dt^2} + \frac{1}{RC} \frac{di_L}{dt} + \frac{1}{LC} i_L = I_0 \quad (۲۲.۲)$$

جواب کامل این معادله دیفرانسیل برابر جواب همگن ^{۳۸} (که در بخش ۹.۲ بدست آمد) بعلاوه جواب خصوصی ^{۳۹} آن خواهد بود.

بیاید فرض کنیم که جریان تحریک یک جریان سینوزوئیدال بصورت $i_p = A_p \sin \omega_0 t$ است. برای بدست آوردن جواب خصوصی ^{۴۰} ما جوابی را در نظر می‌گیریم که جمع مشتق اول و دوم آن را بتوان برابر فرمی از جریان تحریک فرض کرد. بنابراین، فرض کنید

$$i_p = A_p \sin \omega_0 t + B_p \cos \omega_0 t \quad (۲۳.۲)$$

که در آن A_p و B_p دو مقدار ثابت هستند. در این صورت،

$$\frac{di_p}{dt} = \omega_0 (A_p \cos \omega_0 t - B_p \sin \omega_0 t)$$

و

$$\frac{d^2 i_p}{dt^2} = \omega_0^2 (-A_p \sin \omega_0 t - B_p \cos \omega_0 t)$$

با قرار دادن این روابط در معادله ^(۲۲.۲)، خواهیم داشت

$$(-A_p \omega_0^2 - \frac{1}{RC} \omega_0 B_p + \frac{1}{LC} A_p) \sin \omega_0 t + (-B_p \omega_0^2 + \frac{1}{RC} \omega_0 A_p + \frac{1}{LC} B_p) \cos \omega_0 t = A_0 \sin \omega_0 t \quad (۲۴.۲)$$

با جمع کردن ضرایب جملات سینوس و کسینوس در سمت چپ معادله ^(۲۴.۲) و مساوی قرار دادن آنها با ضرایب جملات سینوس و کسینوس سمت راست، بدست می‌آید؛

$$\frac{1}{LC} - \omega_0^2 A_p - \frac{\omega_0}{RC} B_p = A_0$$

$$\frac{\omega_0}{RC} A_p + \frac{1}{LC} - \omega_0^2 B_p = 0$$

از حل این دو معادله A_p و B_p بدست می‌آید

$$A_p = A_0 \times \frac{\frac{1}{LC} - \omega_0^2}{(\frac{\omega_0}{RC})^2 + (\frac{1}{LC} - \omega_0^2)^2} \quad (۲۵.۲)$$

$$B_p = -A_0 \times \frac{\frac{\omega_0}{RC}}{(\frac{\omega_0}{RC})^2 + (\frac{1}{LC} - \omega_0^2)^2} \quad (۲۶.۲)$$

با قرار دادن روابط ^(۲۵.۲) و ^(۲۶.۲) در رابطه ^(۲۳.۲)، خواهیم داشت؛

$$i_p = A_0 \times \frac{1}{(\frac{\omega_0}{RC})^2 + (\frac{1}{LC} - \omega_0^2)^2} \left(\frac{1}{LC} - \omega_0^2 \right) \sin \omega_0 t - \frac{\omega_0}{RC} \cos \omega_0 t \quad (۲۷.۲)$$

میتوانیم رابطه ^(۲۷.۲) را با استفاده از اتحادهای مثلثاتی بصورت زیر بازنویسی نمائیم

$$\alpha \sin \omega t + \beta \cos \omega t = \gamma \sin(\omega t - \phi)$$

که در آن

$$\gamma = \sqrt{\alpha^2 + \beta^2}$$

^{۳۸}homogeneous solution

^{۳۹}particular solution

$$\phi = \tan^{-1} \frac{\beta}{\alpha}$$

با قرار دادن این روابط در معادله (۲۷.۲)، داریم

$$i_p = A_0 \times \frac{1}{\sqrt{(\frac{\omega_0}{RC})^2 + (\frac{1}{LC} - \omega_0^2)^2}} \sin(\omega_0 t - \phi) \quad (28.2)$$

با استفاده از تعاریف پروژه ۱۰.۲ برای فرکانس طبیعی، $\omega_n = \sqrt{\frac{1}{LC}}$ و ضریب میرایی $\zeta =$

$$\text{میتوانیم با عملیات جبری معادله (۲۸.۲) بصورت زیر بنویسیم}$$

$$i_p = \frac{A_0}{\omega_n} \frac{1}{\sqrt{(1 - \frac{\omega_0}{\omega_n})^2 + (2\zeta \frac{\omega_0}{\omega_n})^2}} \sin(\omega_0 t - \phi) \quad (29.2)$$

جمله

$$\frac{A_0}{\omega_n} \frac{1}{\sqrt{(1 - \frac{\omega_0}{\omega_n})^2 + (2\zeta \frac{\omega_0}{\omega_n})^2}}$$

دامنه نوسان است. برای مقدار $\frac{A_0}{\omega_n}$ داده شده، هر قدر جمله $\frac{1}{\sqrt{(1 - \frac{\omega_0}{\omega_n})^2 + (2\zeta \frac{\omega_0}{\omega_n})^2}}$ بزرگتر باشد،

دامنه بیشتر میشود. فرض کنید دامنه برابر

$$\frac{1}{\sqrt{(1 - \frac{\omega_0}{\omega_n})^2 + (2\zeta \frac{\omega_0}{\omega_n})^2}}$$

باشد،

۱. برنامه‌ای بزبان متلب نوشته بطوری که دامنه را بر حسب $\frac{\omega_0}{\omega_n}$ بازاء مقادیر

$\zeta = 1.0, 0.5, 0.25, 0.10, 0.05$ در قدمهای ۰.۰۱ رسم کند.

۲. اگر $\omega_0 \rightarrow \omega_n$ چه اتفاق می‌افتد؟

۱۲.۲ پروژه ۲۵.۲ را بصورت زیر اصلاح کنید:

۱. ازتابع خود نوشته که درون‌یابی را انجام میدهد و بصورت فایل *m*. (ام‌فایل) ذخیره شده است استفاده کنید. فرمول درون‌یابی برای $y(x) = f(x)$ بصورت زیر است:

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} \times (y_2 - y_1)$$

که در آن نقاط (x_1, x_2) نزدیکترین نقاط به x و مقادیر y بترتیب مربوط به (x_1, x_2) هاستند. از این بیان کلی استفاده کرده و برای T_1, T_2, μ_1, μ_2 بصورت تابع خودنوشته استفاده کنید. در اینجا T_1, T_2 نزدیکترین مقادیر جدول دما به درجه حرارتی است که μ تعیین می‌شود و μ_1, μ_2 مقادیر μ بترتیب در T_1, T_2 هستند.

۲. جدول داده‌ها را در روی صفحه نمایش چاپ کرده، سپس از کاربر بخواهید که اگر درون‌یابی برای μ در سایر دماهایی که در جدول نیست می‌خواهد، ارائه دهد. اگر جواب مثبت است، از کاربر بخواهید که دما را وارد کندو درون‌یابی انجام و نتیجه چاپ شود. از گره کلید برای تعیین خواستن درون‌یابی استفاده کنید. اگر جواب منفی است، از برنامه خارج شوید.

۳. اگر در بخش دوم (قبل) جواب مثبت است، سپس از کاربر بخواهید، اگر می‌خواهد دمای دمای دیگری بدهد، اگر جواب مثبت است، از کاربر بخواهید که دما را وارد کند، و درون‌یابی را انجام داده و جواب را چاپ کنید. این فرآیند را تاموقوعی که جواب منفی است ادامه دهید. کاربر دماهای زیر را هر بار بر حسب (k) وارد می‌کند: ۱۲۵، ۱۸۰، ۳۳۴، و ۴۲۷.

۱۳.۲ پروژه

این پروژه اصلاح مثال ۲۴.۲ است که "تابع ناشناس" درون‌یابی شامل x, x_1, x_2, y_1, y_2 را با فرمول درون‌یابی زیر انجام می‌دهد.

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} \times (y_2 - y_1)$$

در اینجا نقاط (x_1, x_2) نزدیکترین نقاط به x و مقادیر y_1, y_2 مقادیر y بترتیب مربوط به (x_1, x_2) هاستند. در مثال ۶.۲، نشان داده شد که متغیر ورودی به تابع لازم نیست همان‌نامی که در برنامه بکار برده شد را داشته باشد. برای تطابق یک یک، موقعیت آرگومان در خواندن برنامه باید در همان موقعیت در تابع باشد. مقادیر جدول برای تحرک الکترون μ بر حسب دمای T چنین است:

$$T = [100 \ 150 \ 200 \ 250 \ 300 \ 350 \ 400 \ 450]$$

$$\mu = [19650 \ 7427 \ 3723 \ 2180 \ 1407 \ 972.0 \ 705.5 \ 531.8]$$

واحد T بر حسب (K) و واحد μ برابر ($cm^2/V-s$) است.

۱. برنامه‌ای بزبان متلب نوشه بطوری که تابع ناشناس داشته و تحرک الکترون را در دماهای زیر TT درون‌یابی کند:

$$TT = [110 \ 165 \ 215 \ 365 \ 440]$$

۲. جدولی برای مقادیر μ در دماهای TT چاپ کنید.

۱۴.۲ پروژه

در پردازش سیگنال‌های گستته، سیگنال‌ها، بصورت رشتہ‌ای از مقادیر عددی گستته بوده و در مقابل آن شکل موج پیوسته، پردازش سیگنال زمان پیوسته است.

شکل (۱۹.۲)(الف) یک شکل موج زمان پیوسته (سینوسی یک کیلوهرتز) همراه سیگنال زمان گستته خود که با سرعت نمونه‌برداری ۸۰۰۰ نمونه در ثانیه (برابر با ۱۲۵ میکروثانیه) نمونه‌برداری شده، را نشان میدهد. این سرعت نمونه‌برداری معمولاً در درستگاه‌های تلفنی بکار برده می‌شود. برنامه‌ای بزبان متلب نوشه که

۱. مقادیر شکل موج را با سرعت ۸۰۰۰ نمونه در ثانیه در یک دوره $msec$ $t \leq 2 \leq 0$ برای ۱۷ نمونه محاسبه کند. فرض کنید مقادیر بیشینه و کمینه موج سینوسی 1 ± 1 است.

۲. جدولی تدوین کنید که شامل تعداد نمونه‌ها، زمان نمونه‌ها، و مقادیر نمونه‌ها باشد.

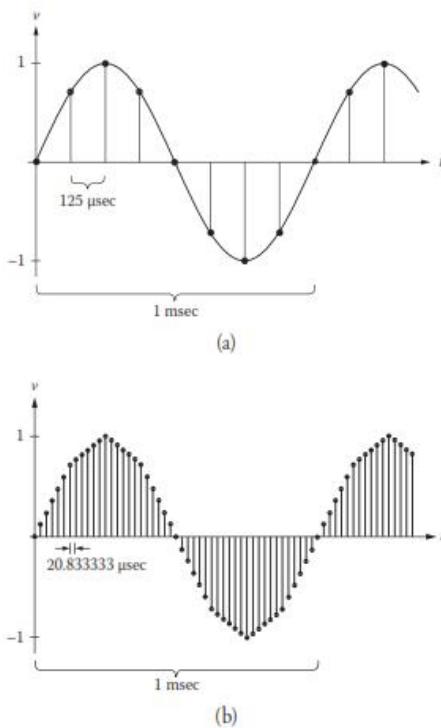
۳. با استفاده از تابع $interp1$ متلب، شکل موج نمونه‌برداری بالا (یعنی در سرعت ۴۸,۰۰۰ نمونه در ثانیه) انجام دهد. برای دوری از خطای گرد کردن، $\Delta t = 20.833333\mu sec$ در نظر گیرید. باید با مقادیر ۹۷ اندازه نمونه خاتمه باید.

۴. شکل موج نمونه‌برداری شده و اصلی را روی یک نمودار رسم کنید. برای شکل موج اصلی، از دایره و برای شکل موج نمونه‌برداری شده از X برای نمایش نقاط داده استفاده کنید.

۱۵.۲ پروژه

این پروژه شامل تعیین کیفیت شکل موج پروژه شماره ۱۴.۲ است. این با مقایسه شکل موج نمونه‌برداری بالا با یک موج سینوسی یک کیلوهرتز انجام می‌شود. اجرای مرسوم برای سیستم‌های صوتی اعوجاج هارمونیکی کل (THD)^۴ است، که آن اندازه گیری نویز در فرکانس‌های هارمونیکی بعییر از فرکانس مطلوب (در حالت ما یک کیلوهرتز) است. چون سیگنال ما متناوب و دارای فرکانس معلومی است، فرض می‌توان فرض کرد که تمام نواقص در THD شرکت دارند. برنامه‌ای بزبان متلب بنویسید که

^۴. total harmonic distortion (THD)



شکل ۱۹.۲: (الف) یک موج سینوسی یک کیلوهرتز که با سرعت ۸۰۰۰ نمونه در ثانیه نمونهبرداری شده است.
(b) اگر بخواهیم این موج سینوسی را در سرعت نمونهبرداری بالا ۴۸,۰۰۰ نمونه در ثانیه نمونهبرداری کنیم، لازم است پنج نمونه اضافی بین نقاط شکل موج اصلی درون یابی کنیم.

۱. مقادیر شکل موج ایدهآل نمونهبرداری بالا در سرعت ۴۸,۰۰۰ نمونه در ثانیه را در زمان یک دوره $t \leq 0$ ، برای کل ۹۷ نمونه محاسبه کنید. فرض کنید بیشینه و کمینه دامنه ± 1 برای موج سینوسی است.
۲. این شکل موج ایدهآل را از شکل موج نمونهبرداری بالا در سرعت ۴۸,۰۰۰ نمونه در ثانیه کم کرده و شکل موج خطأ V_{error} را با روش درون یابی که در پروژه ۱۴.۲ بکار برده شد محاسبه کنید.
۳. اعوجاج هارمونیکی کل THD را برای شکل موج نمونهبرداری بالا بصورت زیر محاسبه کنید:

$$THD = \sqrt{\frac{(V_{error,RMS})^2}{(V_{ideal,RMS})^2}} \quad (30.2)$$

با استفاده از روش تشریح شده در پروژه ۶.۲ و $(V_{ideal,RMS})$ و $(V_{error,RMS})$ را محاسبه کنید.

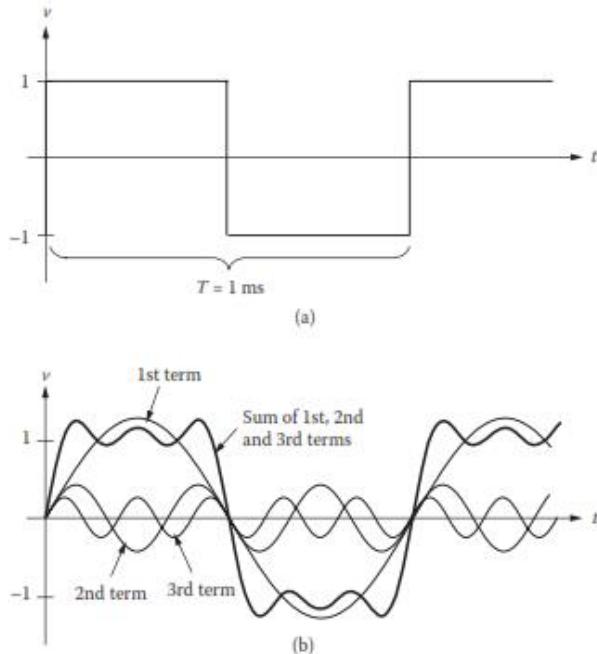
۴. $V_{ideal,RMS}$ و $V_{error,RMS}$ را چاپ کنید.

- ۱۶.۲ پروژه ریاضیات ژوف فوریه میگوید که هر شکل ۴ موج متناوبی را میتوان بر حسب جمع از جملات سینوسی و کسینوسی بیان کرد. برای مثال، موج مربعی شکل (۲۰.۲)(الف) را میتوان بصورت

زیر نوشت:

$$v(t) = \frac{4}{\pi} \sin \frac{2\pi t}{T} + \frac{4}{3\pi} \sin \frac{6\pi t}{T} + \frac{4}{5\pi} \sin \frac{10\pi t}{T} + \dots$$

شکل (۲۰.۲)(ب) مجموع سه جمله اول این سری را نشان میدهد.



شکل ۲۰.۲: (الف) موج مربعی. (ب) سه جمله از سری فوریه موج مربعی.

۱. برنامه‌ای بزبان متلب نوشته بطوری که در آن از تابع خودنوشته `i = sqwave(n, T)` با آرگونهای زیر استفاده شده است.

n تعداد جمله‌ها در سری فوریه.

T زمان تناوب موج مربعی بر حسب ثانیه.

i تعداد نمونه‌ها در یک تناوب.

تابع باید دو آرایه t و V را که هر کدام شامل i هستند بما بدهد بطوری که

$t = i$ یک آرایه از i نقطه زمانی.

$V = i$ یک آرایه از i تقریباً n موج مربعی مقادیر محاسبه شده.

۲. تابع `i = sqwave(n, T)` را اجرا کرده و نتایج را برای مقادیر زیر رسم نمایید.

$T = 1 \text{ msec}$, $i = 1001$, $n = 1, 3, 10, 100$

۳. برای $n = 100$, جدولی از (t, V) در هر بیست نقطه زمانی چاپ کنید.

۱۷.۲ پروژه

آقای باب برای مغازه سخت‌افزارش میخواهد برنامه‌ای داشته باشد که فهرست اموال خود را در مغازش بروز داشته باشد. برنامه‌ای بزبان متلب نوشته برای این هدف بنویسید. برنامه شامل اقلام زیر است: یک فایل داده، یک برنامه متنی اصلی، و یک تابع صدور صورت حساب.

فایل داده:

فایل داده شامل توصیف فهرست اقلام برای فروش، قیمت، و تعداد موجود برای فروش است. این لیست باید لااقل شامل ده قلم جنس باشد. این فایل داده را برنامه متن اصلی میخواند. وقتی جنسی خریداری شد، موجودی فهرست اقلام باید در فایل داده بروز شود.

برنامه متن اصلی:

۱. برنامه متن اصلی اقلام آماده برای فروش، قیمت، و تعداد موجود را در روی صفحه نمایشکر چاپ می‌کند.
۲. در برنامه متن اصلی از کاربر سوال میکند که میخواهد خرید کند و اگر جواب مثبت بود، از کاربر بخواهد ردیف عدد و تعداد جنسی را می‌خواهد خرید کند را بدهد.
۳. اگر کاربر نمی‌خواهد خرید کند، از برنامه خارج شود. اگر کاربر جواب صحیح نداد، روی صفحه نمایشگر پیام خطلا را چاپ کرده و از کاربر بخواهد دوبار برنامه را اجرا نماید. در این صورت از برنامه خارج شود.
۴. اگر

فصل ۳

ماتریس‌ها

۱.۳ مقدمه

در مهندسی، ما اغلب با ماتریس‌ها یا مسائلی که شامل مجموعه‌ای از معادلات خطی است، مواجه می‌شویم. بعلاوه، سنگ بنای مطلب ماتریس است، و بنابراین لازم است فصلی را به ماتریس‌ها اختصاص دهیم. ابتدا، مفاهیم اولیه و عملکرد، و رفتار ماتریس را در مطلب که شامل چندین برنامه مطلب نمونه شامل ماتریس‌ها است را بحث می‌کنیم. سپس بحث را با چندین روش برای حل دستگاه‌های معادلات خطی، شامل استفاده ازتابع inv مطلب دنبال می‌نمائیم. یک مثال از استفاده از ماتریس‌ها در حل مسئله مدارهای مقاومتی را بیان میداریم. در آخر، مسئله مقدار ویژه شامل کاربرد آن در مسئله مدار رزونانسی (مدار تشدید) را بحث می‌کنیم.

۲.۳ عملیات ماتریسی

■ یک آرایه از اعداد بصورت زیر را ماتریس می‌نامند.

$$\mathbf{A} = \begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{matrix}$$

■ یک ماتریس با m سطر و یک ستون را یک بردار ستونی می‌نامند.

■ یک ماتریس یک سطر و n ستونی را یک بردار سطروی می‌نامند.

■ ماتریس‌ها از قوانین جمع، تفریق، و ضرب پیروی می‌کنند.

■ جمع و تفریق: اگر ماتریس A و B دارای تعداد یکسانی سطر و ستون باشند، در این صورت

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{matrix} (a_{11} + b_{11}) & (a_{12} + b_{12}) & \cdots & (a_{1n} + b_{1n}) \\ (a_{21} + b_{21}) & (a_{22} + b_{22}) & \cdots & (a_{2n} + b_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{m1} + b_{m1}) & (a_{m2} + b_{m2}) & \cdots & (a_{mn} + b_{mn}) \end{matrix}$$

$$\mathbf{C} = \mathbf{A} - \mathbf{B} = \begin{pmatrix} (a_{11} - b_{11}) & (a_{12} - b_{12}) & \cdots & (a_{1n} - b_{1n}) \\ (a_{21} - b_{21}) & (a_{22} - b_{22}) & \cdots & (a_{2n} - b_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{m1} - b_{m1}) & (a_{m2} - b_{m2}) & \cdots & (a_{mn} - b_{mn}) \end{pmatrix}$$

■ جمع و تفریق ماتریس‌های \mathbf{A} و \mathbf{B} فقط وقتی تعریف می‌شوند که \mathbf{A} و \mathbf{B} دارای تعداد سطر و ستون‌های یکسانی باشند.

■ ضرب: حاصل ضرب \mathbf{AB} فقط وقتی تعریف می‌شود که تعداد ستون‌های ماتریس \mathbf{A} برابر تعداد سطرهای ماتریس \mathbf{B} باشد. اگر $\mathbf{C} = \mathbf{AB}$ در این صورت

$$\mathbf{A} = \begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{matrix} \quad \text{و} \quad \mathbf{B} = \begin{matrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{matrix}$$

در این صورت

$$\mathbf{C} = \begin{pmatrix} (a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}) & (a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}) \\ (a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31}) & (a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32}) \end{pmatrix}$$

■ اگر \mathbf{A} دارای m سطر و \mathbf{B} دارای K ستون باشد، در این صورت، $\mathbf{C} = \mathbf{AB}$ دارای m و K ستون خواهد بود. رابطه کلی برای عنصر c_{ij} برابر است با

$$c_{ij} = \sum_{k=1}^K a_{ik}b_{kj}$$

در مطلب، ضرب ماتریس \mathbf{A} و \mathbf{B} را چنین $\mathbf{A} * \mathbf{B}$ وارد می‌کنند.

■ ترانهاد^۱: ترانهاد یک ماتریس عمل تعویض سطرهای و ستونها است. اگر

$$\mathbf{A} = \begin{matrix} 2 & 5 & 1 \\ 7 & 3 & 8 \\ 4 & 5 & 21 \\ 16 & 3 & 10 \end{matrix}$$

در این صورت

$$\mathbf{A}^T = \begin{matrix} 2 & 7 & 4 & 16 \\ 5 & 3 & 5 & 3 \\ 1 & 8 & 21 & 10 \end{matrix}$$

در مطلب، ترانهاد ماتریس \mathbf{A} را بصورت $'\mathbf{A}$ وارد می‌کنند.

■ جمع عناصر یک بردار یا ستون‌های یک ماتریس:
اگر $\mathbf{A} = [a_1 \ a_2 \ a_3]$ باشد، در این صورت، جمع $(\mathbf{A}) = a_1 + a_2 + a_3$ اگر

$$\mathbf{B} = \begin{matrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{matrix}$$

^۱Transpose

بنابراین،

$$\text{جمع } \mathbf{B} = [(b_{11} + b_{21} + b_{31}) \ (b_{12} + b_{22} + b_{32}) \ (b_{13} + b_{23} + b_{33})]$$

در مطلب، جمع ماتریس \mathbf{A} را $\text{sum}(\mathbf{A})$ وارد می‌کنند.

■ ضرب داخلی^۲: ضرب داخلی دو بردار چنین تعریف می‌شود:

$$\mathbf{A} \cdot \mathbf{B} = \sum a_i b_i$$

مثلاً، اگر $\mathbf{B} = [-2 \ 5 \ 2]$ باشد، در این صورت $\mathbf{A} = [4 \ -1 \ 3]$ و $\mathbf{A} \cdot \mathbf{B} = (4)(-2) + (-1)(5) + (3)(2) = -7$

نتیجه حاصل ضرب داخلی همواره یک کمیت اسکالر (یعنی یک عدد) است.

■ ماتریس همانی^۳: ماتریس همانی \mathbf{I} ماتریسی است که عناصر قطر اصلی آن یک و سایر عناصر صفر هستند.

$$\mathbf{I}\mathbf{A} = \mathbf{A}\mathbf{I} = \mathbf{A}$$

برای مثال، برای ماتریس سه در سه \mathbf{A} ،

$$\begin{aligned} \mathbf{A}\mathbf{I} &= \begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array} \\ &= \begin{array}{ccc|ccc} (a_{11} + 0 + 0) & (0 + a_{12} + 0) & (0 + 0 + a_{13}) & 1 & 0 & 0 \\ (a_{21} + 0 + 0) & (0 + a_{22} + 0) & (0 + 0 + a_{23}) & 0 & 1 & 0 \\ (a_{31} + 0 + 0) & (0 + a_{32} + 0) & (0 + 0 + a_{33}) & 0 & 0 & 1 \end{array} \end{aligned}$$

در مطلب، میتوانید ماتریس همانی $n \times n$ را با `eye(n)` تولید کنید.

■ ماتریس معکوس: عکس ماتریس \mathbf{A} را با \mathbf{A}^{-1} نشان میدهد و چنین تعریف می‌شود:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \quad 3 \times 3$$

در مطلب، عکس ماتریس \mathbf{A} را بصورت $\text{inv}(\mathbf{A})$ وارد می‌کنند.

■ دترمینان:^۴

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

در مطلب، دترمینان ماتریس \mathbf{A} بصورت $\det(\mathbf{A})$ وارد می‌کنند.

■ ابعاد^۵: میتوانید اندازه ماتریس \mathbf{A} را با اجرای `size(A)` بدست آورید. این فرمان وقتی مفید است که پیام خطای مانند "ابعاد ماتریس زیاد است" دریافت می‌کنید. با اضافه کردن فرمان `size()` در متن بشمای کمک می‌کند که مسئله را تعیین نماید.

^۱Dot product

^۲Identity

^۳Determinant

^۴Dimensions

■ عملیات عنصر به عنصر: دو ماتریس با ابعاد یکسان داده شده است، می‌توانیم ضرب و تقسیم عنصر در عنصر در متلب با $*$ و $/$. انجام دهیم.
اگر

$$\mathbf{A} = \begin{matrix} a_1 & a_2 & a_3 \end{matrix} \quad \text{و} \quad \mathbf{B} = \begin{matrix} b_1 & b_2 & b_3 \end{matrix}$$

در این صورت

$$\mathbf{C} = \mathbf{A} \cdot * \mathbf{B} = \begin{matrix} a_1 b_1 & a_2 b_2 & a_3 b_3 \end{matrix}$$

$$\mathbf{D} = \mathbf{A} \cdot / \mathbf{B} = \begin{matrix} \frac{a_1}{b_1} & \frac{a_2}{b_2} & \frac{a_3}{b_3} \end{matrix}$$

ضمناً، توجه کنید که ضرب داخلی را می‌توان بصورت ترکیبی از ضرب عنصر به عنصر و جمع بکار برد:

$$\mathbf{A} \cdot \mathbf{B} = \text{sum}(\mathbf{A} \cdot * \mathbf{B}) = \text{sum}\left(\frac{a_1}{b_1} \quad \frac{a_2}{b_2} \quad \frac{a_3}{b_3}\right) = a_1 b_1 + a_2 b_2 + a_3 b_3$$

■ ضرب دو تابع برداری:

بطوری که در فصل دوم تشریح شد، متلب بما این اجازه را میدهد که یک تابع را بر یک بردار انجام داده و نتیجه یک بردار شود. البته، اگر متنی لازم داشته باشد که عملیات ریاضی روی دو تابع (از قبیل ضرب دو تابع برداری) انجام دهد، در آن صورت عملیات عنصر در عنصر لازم خواهد داشت.

در مثال زیر، ما ضرب دو تابع برداری را از دو روش مستقیم و غیرمستقیم با استفاده از حلقه for و ضرب عناصر توابع محاسبه می‌کنیم. این متن ساده را در پنجره متن نوسته و تحت عنوان آنرا ذخیره و اجرا نمائید.

```
% matrix_example.m
clear; clc;
x = 0:30:360;
% y1 is the product of two vector functions
y1 = sind(x).* cosd(x);
fprintf('x    y1    y2\n');
fprintf('-----\n');
for n = 1:length(x);
% y2(n) is the product of the elements of the two
% functions.
y2(n) = sind(x(n)) * cosd(x(n));
fprintf('%5.1f %8.5f %8.5f \n',x(n),y1(n),y2(n));
end
%-----
```

مثال ۱.۳: عملیات ماتریسی

در مثال زیر چندین عملیات ماتریسی را شامل جمع، ضرب داخلی، ضرب عنصر در عنصر، تقسیم، جمع، و ترانهاد انجام میدهیم.

```
% Example_3_1.m
% This program demonstrates matrix algebra in MATLAB
clear; clc;
```

```

a=[1 5 9]
b=[2 6 12]
c=a+b
d=dot(a,b)
e=a.*b
f=a./b
g=sum(a.*b)
h=a*b'
%-----

```

۳.۳ دستگاه معادلات خطی

در مهندسی، ما اغلب با حل مسائل مجموعه‌ای از معادلات خطی مواجه می‌شویم.
مجموعه‌ای از معادلات خطی داده شده است:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= c_1 \\
 a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= c_2 \\
 &\vdots &=& \vdots \\
 a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= c_n
 \end{aligned} \tag{۱.۳}$$

این مجموعه‌ای شامل n معادله و n مجھول است. راحت‌تر است که این مجموعه را بصورت معادله ماتریسی بنویسیم

$$\mathbf{AX} = \mathbf{C} \tag{۲.۳}$$

که در آن

$$\mathbf{A} = \begin{matrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{matrix}, \quad \mathbf{X} = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix}, \quad \mathbf{C} = \begin{matrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{matrix}$$

ماتریس \mathbf{A} دارای n سطر و n ستون است.

ماتریس \mathbf{X} دارای n سطر و یک ستون است.

توجه: تعداد ستون‌های \mathbf{A} باید برابر تعداد سطرهای \mathbf{X} باشد. در غیراین صورت، ضرب تعریف نمی‌شود.

ماتریس \mathbf{C} دارای n سطر و یک ستون است.

در جبر ماتریسی، \mathbf{X} را می‌توان از ضرب دوطرف در \mathbf{A}^{-1} بدست آورد، یعنی،

$$\underbrace{\mathbf{A}^{-1}\mathbf{A}}_{\mathbf{I}} \mathbf{X} = \mathbf{A}^{-1}\mathbf{C} \quad \mathbf{IX} = \mathbf{X} = \mathbf{A}^{-1}\mathbf{C}$$

متلب توابع زیر را برای حل یک دستگاه معادلات خطی پیشنهاد می‌کند:

تابع inv ■

برای حل یک دستگاه معادلات خطی در متلب، میتوان نوشت

$$X = \text{inv}(A) * C$$

این روش حل یک دستگاه معادلات خطی با استفاده از تابع inv خیلی از روش حذف گوسی که در مطلب بعدی خواهیم گفت از نظر ریاضی پیچیده‌تر است.

تابع حذف گوسی:

متلب برای حل یک دستگاه معادلات خطی، $\mathbf{AX} = \mathbf{C}$ ، بوسیله حذف گوسی چنین می‌نویسد

$$\mathbf{X} = \mathbf{A} \backslash \mathbf{C}$$

توجه کنید که استفاده از عملگر \ برای حل \mathbf{X} با حذف گوسی است.

مثال ۲.۳:

مثال زیر سه معادله دستگاه خطی و نوشتن آنرا در یک فایل نشان می‌دهد:

$$\begin{aligned} 3x_1 + 2x_2 - x_3 &= 10 \\ -x_1 + 3x_2 + 2x_3 &= 5 \\ x_1 - x_2 - x_3 &= -1 \end{aligned}$$

```
% Example_3_2.m
% This program solves a simple linear system of equations
% by both the use of MATLAB's inverse matrix function and
% by the Gauss elimination method.
clc; clear;
A=[3 2 -1; -1 3 2; 1 -1 -1]
C=[10 5 -1] '
% X1 is the solution using matrix inverse function:
X1=inv(A)*C
% X2 is the solution using Gauss elimination method:
X2=A\C
% check solution:
A*X1
% Use the size() command to determine the number of rows
% and the number of columns in matrix A.
[A_rows A_cols] = size(A)
% Print matrix A, matrix C and the solution of the linear
% system of equations to the file output.txt:
fid=fopen('output.txt','w');
fprintf(fid,'A matrix:\n');
for i=1:A_rows
for j=1:A_cols
fprintf(fid,'%6.1f',A(i,j));
end
fprintf(fid,'\n');
end
fprintf(fid,'C vector:\n');
for i=1:length(C)
fprintf(fid,'%6.1f',C(i));
end
fprintf(fid,'\n');
fprintf(fid,'Solution X1 (using inverse matrix):\n');
for i=1:length(X1)
fprintf(fid,'%6.1f',X1(i));
end
```

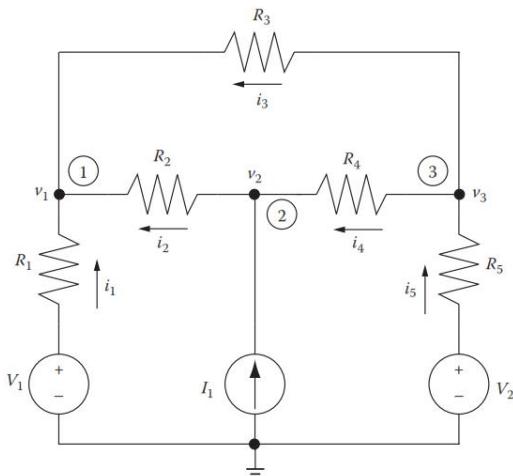
```

end
fprintf(fid, '\n');
fprintf(fid, 'Solution X2 (using Gauss elimination):\n');
for i=1:length(X2)
fprintf(fid, '%6.1f', X2(i));
end
fprintf(fid, '\n');
fclose(fid);
%-----

```

مثال ۳.۳: مثالی از مدارات مقاومتی

نمونه‌ای از کاربرد معادلات خطی را می‌توان در مدار مقاومتی شکل (۱.۳) مشاهده کرد. هدف اینجا پیدا کردن ولتاژ گره‌های v_1 , v_2 , و v_3 بر حسب توابعی از ولتاژهای ورودی V_1 و V_2 و جریان ورودی I_1 است.



شکل ۱.۳: پنجره متلب روی میزکار

با استفاده از قانون اهم ($v = iR$) و قانون جریانهای کیرشهف (KCL), می‌توانیم رابطه‌هایی برای جمع جریانهای در هر گره بنویسیم. (KCL) بما می‌گوید که کل جریان وارد شونده در هر گره مدار باید صفر باشد. در شکل (۱.۳)، جریان مقاومت‌ها را بصورت i_n نوشته و جهت آنها را بطور دلخواه مانند شکل در نظر می‌گیریم. با نوشتن سه معادله (KCL) برای جریانهای وارد شونده در هر گره (با توجه به این که جریانهای خارج شونده از گره را از نظر ریاضی با علامت منفی در نظر گیریم) خواهیم داشت: در گره شماره یک:

$$i_1 + i_2 + i_3 = 0 \quad (3.3)$$

$$i_1 - i_2 + i_4 = 0 \quad (4.3)$$

$$-i_3 - i_4 + i_5 = 0 \quad (5.3)$$

اکنون این سه معادله را با استفاده از قانون اهم بر حسب ولتاژها بازنویسی می‌کنیم. توجه کنید که در ($G_n = 1/R_n$ (یعنی G_n این نوع مسائل، اغلب ساده‌تر است آنها را بجای مقاومت، بر حسب هدایت G_n (یعنی

نوشته و از قانون اهم استفاده نمایم. بنابراین،

$$i_1 = \frac{V_1 - v_1}{R_1} \rightarrow i_1 = (V_1 - v_1)G_1 \quad (6.3)$$

$$i_2 = \frac{v_2 - v_1}{R_2} \rightarrow i_2 = (v_2 - v_1)G_2 \quad (7.3)$$

$$i_3 = \frac{v_3 - v_1}{R_3} \rightarrow i_3 = (v_3 - v_1)G_3 \quad (8.3)$$

$$i_4 = \frac{v_3 - v_2}{R_4} \rightarrow i_4 = (v_3 - v_2)G_4 \quad (9.3)$$

$$i_5 = \frac{V_2 - v_3}{R_5} \rightarrow i_5 = (V_2 - v_3)G_5 \quad (10.3)$$

با قرار دادن این روابط برای i_1 تا i_5 در معادلات ۳.۳ تا ۵.۳ خواهیم داشت
 $(V_1 - v_1)G_1 + (v_2 - v_1)G_2 + (v_3 - v_1)G_3 = 0 \quad (11.3)$

$$I_1 - (v_2 - v_1)G_2 + (v_3 - v_2)G_4 = 0 \quad (12.3)$$

$$-(v_3 - v_1)G_3 - (v_3 - v_2)G_4 + (V_2 - v_3)G_5 = 0 \quad (13.3)$$

این روابط را می‌توان بصورت یک دستگاه سه معادله با سه مجهول نوشت:

$$\begin{matrix} (G_1 + G_2 + G_3) & -G_2 & -G_3 & v_1 & V_1 G_1 \\ -G_2 & (G_2 + G_4) & -G_4 & v_2 & I_1 \\ -G_3 & -G_4 & (G_3 + G_4 + G_5) & v_3 & V_2 G_5 \end{matrix} = \begin{matrix} & & & & \\ & & & & \\ & & & & \end{matrix} \quad (14.3)$$

این معادله بصورت $AX = C$ است و می‌توان همانطوری که در مثال ۲.۳ قبل نشان داده شد، با استفاده از روش حذف گوسی حل کرد.

مقادیر زیر را برای ولتاژ گرهها بکار میبریم

$$R_1 = 2200, \quad R_2 = 10k, \quad R_3 = 6900, \quad R_4 = 9100, \quad R_5 = 3300$$

$$V_1 = 12 \text{ V}, \quad V_2 = 3.3 \text{ V}, \quad I_1 = 2 \text{ mA}$$

```
Example_3_3.m Resistive Circuit
% This program solves for the internal node voltages for
% the circuit shown in Figure 3.1.
% The conductances G are in units of siemens
% The node voltage V are in units of volts
% The currents I are in units of amps
clear; clc;
g1 = 1/2200;
g2 = 1/10000;
g3 = 1/6900;
g4 = 1/9100;
g5 = 1/3300;
V1 = 12;
V2 = 3.3;
I1 = .002;
```

```
% Put the problem into standard form AX=C:
A = [ g1+g2+g3 -g2 -g3; -g2 g2+g4 -g4; -g3 -g4 g3+g4+g5];
C = [ V1*g1; I1; V2*g5 ];
X = A \ C ;
% print the results
fprintf('V1=%9.1f V\nV2=%9.1f V\nI1=%9.1e A\n',V1,V2,I1);
fprintf('g1=%9.5f S\n g2=%9.5f S\n g3=%9.5f S\n',g1,g2,g3);
fprintf('g4=%9.5f S\n g5=%9.5f S\n',g4,g5);
fprintf('\n');
fprintf('Node # v (volts) \n');
fprintf('-----\n');
for n=1:length(C)
fprintf('%3i %9.1f\n',n,X(n));
end
%-----
```

۴.۳ حذف گوسی

بطوری که قبلاً بحث شد، روش حذف گوسی از نظر ریاضی بیش از روش ماتریس معکوس یا روش دترمینان با قاعده کرامر^۶ کارآمدتر و بهره‌ور است. می‌خواهیم روش حذف گوسی را برای حل یک دستگاه معادلات خطی که با معادله (۱۲) بیان شد و بصورت معادله (۲۳) نشان داده شد، استفاده نمائیم. در روش حذف گوسی، دستگاه اصلی بیک مجموعه مثلثی معادل تقلیل یافته که می‌تواند بسادگی با جاگذاری معکوس حل شود. مجموعه معادل تقلیل یافته بصورت معادلات زیر است:

$$\begin{aligned} \tilde{a}_{11}x_1 + \tilde{a}_{12}x_2 + \tilde{a}_{13}x_3 + \cdots + \tilde{a}_{1n}x_n &= \tilde{c}_1 \\ \tilde{a}_{22}x_2 + \tilde{a}_{23}x_3 + \cdots + \tilde{a}_{2n}x_n &= \tilde{c}_2 \\ \tilde{a}_{33}x_3 + \cdots + \tilde{a}_{3n}x_n &= \tilde{c}_3 \\ \vdots &= \vdots \\ \tilde{a}_{n-1,n-1}x_{n-1} + \tilde{a}_{n-1,n}x_n &= \tilde{c}_{n-1} \\ \tilde{a}_{n,n}x_n &= \tilde{c}_n \end{aligned} \quad (15.3)$$

که در آن علامت سرکج ~ مجموعه ضرایب جدید است (باید بدست آید) و ماتریس ضرایب جدید $\tilde{\mathbf{A}}$ قطری است (یعنی تمام ضرایب در سمت چپ قطر اصلی صفر هستند). در این صورت

$$\begin{aligned} x_n &= \tilde{c}_n / \tilde{a}_{n,n} \\ x_{n-1} &= \frac{1}{\tilde{a}_{n-1,n-1}} (\tilde{c}_{n-1} - \tilde{a}_{n-1,n}x_n) \\ &\vdots \end{aligned}$$

^۶Cramer's rule

برای تعیین مجموعه معادل تقلیل یافته \tilde{A} و \tilde{C} , راحت‌تر است که ضرایب ماتریس اصلی A را با ماتریس C مطابق فرم زیر افزایش دهیم:

$$\mathbf{A}_{aug} = \begin{array}{cccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & c_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & c_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & c_n \end{array} \quad (16.3)$$

در زیر دستورالعمل تعیین مجموعه معادل تقلیل یافته آورده می‌شود:

۱. اولین سطر معادله (۱.۳) را در a_{21}/a_{11} ضرب کرده و از سطر دوم کم کنید. خواهیم داشت

$$a_{21} = a_{21} - \frac{a_{21}}{a_{11}} \times a_{11}, \quad a_{22} = a_{22} - \frac{a_{21}}{a_{11}} \times a_{12}, \quad a_{23} = a_{23} - \frac{a_{21}}{a_{11}} \times a_{13}, \quad \dots$$

$$c_2 = c_2 - a_{21} - \frac{a_{21}}{a_{11}} \times c_1 \quad 9$$

این $a_{21} = 0$ میکند.

۲. برای سومین سطر: اولین سطر معادله (۱.۳) را در a_{31}/a_{11} ضرب کرده و از سطر سوم کم کنید، میدهد

$$a_{31} = a_{31} - \frac{a_{31}}{a_{11}} \times a_{11}, \quad a_{32} = a_{32} - \frac{a_{31}}{a_{11}} \times a_{12}, \quad a_{33} = a_{33} - \frac{a_{31}}{a_{11}} \times a_{13}, \quad \dots$$

$$c_3 = c_3 - \frac{a_{31}}{a_{11}} \times c_1 \quad 9$$

این $a_{31} = 0$ میکند.

۳. این فرآیند را برای بقیه سطرهای $n, 4, 5, 6, \dots$ تکرار کنید. سطر اول اصلی بصورت اولیه خود باقی می‌ماند. تمام سایر سطرهای اصلاح شده و ضرایب جدیدی با $(')$ مشخص می‌شود. بجر سطر اول، مجموعه شامل x_1 نخواهد بود.

۴. برای مراحل بعدی، سطر اول معادله (۱.۳) بعنوان محور اصلی کار یا سطر پاشنه^v و a_{11} بعنوان عنصر اصلی یا عنصر پاشنه^l استفاده می‌شود. اکنون سطر دوم جدید بعنوان سطر پاشنه استفاده شده و مراحل اول تا سوم برای بقیه سطرهای زیر سطر دوم تکرار می‌شود. بنابراین، ضرب و تکرار سطر دوم جدید در $\frac{a_{32}}{a_{22}}$ ضرب و از سطر سوم کم می‌شود، میدهد

$$a_{32} = a_{32} - \frac{a_{32}}{a_{22}} \times a_{22}, \quad a_{33} = a_{33} - \frac{a_{32}}{a_{22}} \times a_{23}, \quad a_{34} = a_{34} - \frac{a_{32}}{a_{22}} \times a_{24}, \quad \dots$$

$$c_3 = c_3 - \frac{a_{32}}{a_{22}} \times c_2 \quad 9$$

این $a_{32} = 0$ میکند.

بهمنین نحو، سطر دوم جدید را در $\frac{a_{42}}{a_{22}}$ ضرب و از سطر چهارم کم می‌شود، خواهد شد.

$$a_{42} = a_{42} - \frac{a_{42}}{a_{22}} \times a_{22}, \quad a_{43} = a_{43} - \frac{a_{42}}{a_{22}} \times a_{23}, \quad a_{44} = a_{44} - \frac{a_{42}}{a_{22}} \times a_{24}, \quad \dots$$

^vpivot row

^lpivot element

۹

$$c_4 = c_4 - \frac{a_{42}}{a_{22}} \times c_2$$

این $a_{42} = 0$ میکند.

این فرایند برای سایر سطرهای $n, n-1, \dots, 5, 6$ ادامه می‌یابد. بجز برای سطر دوم جدید، مجموعه شامل x_2 نخواهد بود.

۵. اکنون سطر بعدی بعنوان سطر اصلی یا پاشنه استفاده و فرآیند تا سطر $(n-1)$ بعنوان سطر پاشنه ادامه می‌یابد. وقتی این کامل شد، دستگاه جدید مثلثی است و می‌توان با جایگذاری معکوس حل شود.

رابطه کلی برای ضرایب جدید برابر است با

$$a_{ij} = a_{ij} - \frac{a_{ik}}{a_{kk}} \times a_{kj} \quad \text{برای } i = k+1, k+2, \dots, n \quad \text{و } j = k+1, k+2, \dots, n$$

که در آن k سطر پاشنه است.

عملیات فقط روی a_{ij} و c_j اثر می‌کند. بنابراین، فقط لازم است روی ماتریس‌های A و C عمل نمائیم.

مثال ۴.۳:

دستگاه زیر داده شده است

$$\begin{array}{lcl} x_1 - 3x_2 + x_3 & = & -4 \\ -3x_1 + 4x_2 + 3x_3 & = & -10 \\ 2x_1 + 3x_2 - 2x_3 & = & 18 \end{array}$$

ماتریس ضرایب اضافه شده برابر است با:

$$\mathbf{A}_{aug} = \left| \begin{array}{ccc|c} 1 & -3 & 1 & -4 \\ -3 & 4 & 3 & -10 \\ 2 & 3 & -2 & 18 \end{array} \right|$$

سطر اول را در $\frac{a_{21}}{a_{11}} = \frac{-3}{1}$ ضرب کرده و از سطر دوم کم کنیم
 سطر دوم خواهد شد $(-3 + 3), (4 - 9), (3 + 3), (-10 - 12) = (0, -5, 6, -22)$
 ماتریس جدید خواهد شد

$$\mathbf{A}_{aug, Equiv} = \left| \begin{array}{ccc|c} 1 & -3 & 1 & -4 \\ 0 & -5 & 6 & -22 \\ 2 & 3 & -2 & 18 \end{array} \right|$$

با ضرب سطر اول در $\frac{a_{31}}{a_{11}} = \frac{2}{1}$ و کم کردن از سطر سوم
 سطر سوم خواهد شد $(2 - 2), (3 + 6), (-2 - 2), (18 - 8) = (0, 9, -4, 26)$
 ماتریس جدید خواهد شد

$$\mathbf{A}_{aug, Equiv} = \left| \begin{array}{ccc|c} 1 & -3 & 1 & -4 \\ 0 & -5 & 6 & -22 \\ 0 & 9 & -4 & 26 \end{array} \right|$$

اکنون سطر دوم را بعنوان سطر پاشنه استفاده می‌کنیم
 $\mathbf{A}_{aug, Equiv} = \left| \begin{array}{ccc|c} 1 & -3 & 1 & -4 \\ 0 & -5 & 6 & -22 \\ 0 & 9 & -4 & 26 \end{array} \right|$

سطر دوم را در $\frac{a_{32}}{a_{22}} = -\frac{9}{5}$ ضرب کرده و از سطر سوم کم کنیم

$$(0 - 0), 9 - \frac{9}{5} \times 5, -4 + \frac{9}{5} \times 6, 26 - \frac{9}{5} \times 22 = 0, 0, \frac{34}{5}, -\frac{68}{5}$$

سطر سوم خواهد شد

ماتریس جدید برابر است با

$$\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|c} 1 & -3 & 1 & -4 \\ 0 & -5 & 6 & -22 \\ 0 & 0 & \frac{34}{5} & -\frac{68}{5} \end{array}$$

اکنون دستگاه مثلثی است، و دستگاه را می‌توان بصورت زیر نوشت:

$$\mathbf{A}_{Equiv} \mathbf{X} = \mathbf{C}_{Equiv}$$

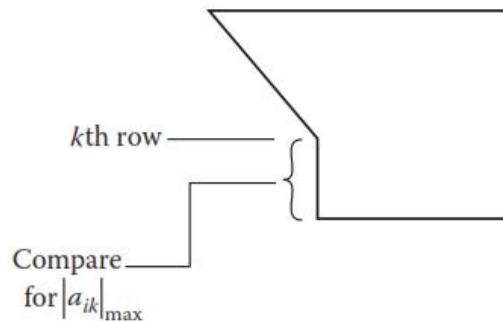
که بفرم زیر خواهد بود

$$\begin{aligned} x_1 - 3x_2 + x_3 &= -4 \\ -5x_2 + 6x_3 &= -22 \\ \frac{34}{5}x_3 &= -\frac{68}{5} \end{aligned}$$

و حل آن خواهد بود:

$$\begin{aligned} \frac{34}{5}x_3 &= -\frac{68}{5} & x_3 &= -2 \\ -5x_2 + 6(-2) &= -22 & x_2 &= 2 \\ x_1 - 3(2) + (-2) &= -4 & x_1 &= 4 \end{aligned}$$

دو نکته مهم در اینجا وجود دارد:



شکل ۲.۳: تعویض سطر

۱. اگر a_{kk} صفر باشد، که در آن k سطر پاشته است، در این صورت فرآیند را نمی‌توان انجام داد.

۲. اگر عنصر پاشته مقدار بیشینه مطلق موجود در مجموعه باشد، دقت بیشتر در حل بدست می‌آید.

یعنی، اگر سطر پاشته k باشد، می‌توان ضرایب a_{ik} را برای $i = k+1, k+2, \dots, n$ مقایسه

کرد (به شکل ۲.۲ مراجعه کنید). اگر $|a_{ik}|_{max} \uparrow |a_{kk}|$. اگر $|a_{ik}|_{max}$ باشد، در این صورت سطر شامل $|a_{ik}|_{max}$ عوض می‌شود. این در ترتیب معادلات اثر می‌کند و در حل اثری ندارد.

اگر بعد از تعویض سطر و یکی از a_{kk} ها باز صفر باقی بماند، در این صورت دستگاه تکین^۹ بوده و جوابی بدست نمی‌آید.

یکی دیگر از ملاحظات: می‌توان نشان داد که اگر مقدار عنصر پاشته خیلی کوچکتر از سایر عناصر ماتریس باشد، استفاده از عنصر پاشته کوچک دقت جواب مسئله را پائین می‌ورد. برای آزمایش این

⁹singular

حالت، می‌توانید معادلات را مقیاس‌بندی کنید، یعنی، هر معادله را به مقدار ضریب بیشینه قدر مطلق در معادله تقسیم کنید. این باعث می‌شود که ضریب بیشینه قدر مطلق برابر $1.0 \ll |a_{kk}|$ شود. اگر $1 \ll |a_{kk}|$ در این صورت جواب احتمالاً جواب دقیقی نخواهد داشت.

۵.۳ روش گوس-جردن

روش گوس-جردن اصلاح روش حذف گوسی است و می‌توان آنرا برای یک دستگاه معادلات بصورت $AX = C$ بکار برد. در این روش، هدف تعیین ماتریس ضرایب معادل بنحوی است که بجز قطر اصلی، تمام عناصر صفر هستند. روش گوس-جردن، با پیدا کردن ماتریس معادل که مثلثی است شروع می‌شود. سپس، بشرط آنکه A یک ماتریس $n \times n$ باشد، با استفاده از سطر i ام بعنوان سطر پاشنه، سطر i ام را در $a_{n-1}/a_{n,n}$ ضرب کرده و از سطر $i-1$ ام کم کرده از اینرو $a_{n-1} = 0$ می‌کنند. این فرآیند برای سطرهای $1, 2, n-3, \dots, n-2$ تکرار می‌شود.

مثال ۵.۳:

بعنوان مثال، روش گوس-جردن بصورت زیر اعمال می‌شود: با ماتریس معادل مثلثی مثال ۴.۳ شروع می‌کنیم،

$$\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|c} 1 & -3 & 1 & -4 \\ 0 & -5 & 6 & -22 \\ 0 & 0 & \frac{34}{5} & -\frac{68}{5} \end{array}$$

سطر سوم را در $a_{23}/a_{33} = 6 \times \frac{5}{34}$ ضرب کرده و از سطر دوم کم می‌کنیم. سطر دوم می‌شود $0 - \frac{30}{34} \times 0, -5 - \frac{30}{34} \times 0, 6 - \frac{30}{34} \times \frac{34}{5}, -22 - \frac{30}{34} \times \frac{68}{5} = (0, -5, 0, -10)$

بعداً، سطر سوم در $a_{13}/a_{33} = \frac{5}{34}$ ضرب و از سطر اول کم می‌کنیم. سطر اول می‌شود $1 - \frac{5}{34} \times 0, -3 - \frac{5}{34} \times 0, 1 - \frac{5}{34} \times \frac{34}{5}, -4 + \frac{5}{34} \times \frac{68}{5} = (1, -3, 0, -2)$

اکنون ماتریس $\mathbf{A}_{aug, Equiv}$ جدید می‌شود

$$\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|c} 1 & -3 & 0 & -2 \\ 0 & -5 & 0 & -10 \\ 0 & 0 & \frac{34}{5} & -\frac{68}{5} \end{array}$$

اکنون، سطر دوم بعنوان سطر پاشنه بکار میرود. سطر دوم را در $a_{12}/a_{22} = \frac{3}{5}$ ضرب کرده و از سطر اول کم می‌کنیم. سطر اول می‌شود،

$$1 - \frac{3}{5} \times 0, -3 - \frac{3}{5} \times 5, 0 - \frac{3}{5} \times 0, -2 + \frac{3}{5} \times 10 = (1, 0, 0, 4)$$

ماتریس $\mathbf{A}_{aug, Equiv}$ جدید می‌شود

$$\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|c} 1 & 0 & 0 & 4 \\ 0 & -5 & 0 & -10 \\ 0 & 0 & \frac{34}{5} & -\frac{68}{5} \end{array}$$

بنابراین، مجموعه معادلات می‌شود

$$\begin{aligned} x_1 &= 4 \\ -5x_2 &= -10 \\ \frac{34}{5} &= -\frac{68}{5} \end{aligned}$$

و در نتیجه $x_1 = 4$ ، $x_2 = 2$ ، $x_3 = -2$ که همان جوابی است که قبلاً با روش حذف گوسی بدست آمد.

۶.۳ تعداد جوابها

فرض کنید یک برنامه حذف گوسی انجام شده و نتایج زیر بدست آمده است:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= c_1 \\ a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= c_2 \\ a_{33}x_3 + \cdots + a_{3n}x_n &= c_3 \\ \vdots &= \vdots \\ a_{rr}x_r + \cdots + a_{rn}x_n &= c_r \\ 0 &= c_{r+1} \\ 0 &= c_{r+2} \\ \vdots & \\ 0 &= c_n \end{aligned}$$

که در آن $r < n$ و $a_{11}, a_{12}, \dots, a_{rr}$ مخالف صفر هستند. دو حالت امکان پذیر است:

۱. اگر یکی از c_{r+1} تا c_n صفر نباشد، جوابی وجود ندارد.
۲. اگر تمام c_{r+1} تا c_n صفر باشند، بی‌نهایت جواب وجود دارد.

اگر $r = n$ و $a_{11}, a_{12}, \dots, a_{nn}$ صفر نباشند، در این صورت دستگاه بصورت زیر خواهد بود:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= c_1 \\ a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= c_2 \\ a_{33}x_3 + \cdots + a_{3n}x_n &= c_3 \\ \vdots &= \vdots \\ a_{nn}x_n &= c_n \end{aligned}$$

برای این حالت فقط یک جواب وجود دارد.

۷.۳ ماتریس معکوس

دستگاه معادلات بصورت کلی زیر داده شده است
 $\mathbf{AX} = \mathbf{C}$

می‌توانیم با ضرب دو طرف این معادله با ماتریس \mathbf{A}^{-1} ، آنرا حل کنیم:
 $\mathbf{A}^{-1}\mathbf{AX} = \mathbf{A}^{-1}\mathbf{C}$

حل به روش متلب بصورت زیر است
 $X = \text{inv}(\mathbf{A}) * \mathbf{C}$ روش معکوس

یا

$X = A/C$ روش حذف گوسی

حال ببینیم در تعیین \mathbf{A}^{-1} چه چیزی در گیر است.

فرض کنید $\mathbf{B} = \mathbf{A}^{-1}$ است. در این صورت $\mathbf{BA} = \mathbf{I}$ است. ما آنرا برای ماتریس 3×3 نشان خواهیم داد:

$$\begin{array}{ccc|ccc|ccc} b_{11} & b_{12} & b_{13} & a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ b_{21} & b_{22} & b_{23} & a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ b_{31} & b_{32} & b_{33} & a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}$$

اولین سطر $\mathbf{B}\mathbf{A}$

$$b_{11}a_{11} + b_{12}a_{21} + b_{13}a_{31} = 1 \quad \text{عنصر} : (1, 1)$$

$$b_{11}a_{12} + b_{12}a_{22} + b_{13}a_{32} = 0 \quad \text{عنصر} : (1, 2)$$

$$b_{11}a_{13} + b_{12}a_{23} + b_{13}a_{33} = 0 \quad \text{عنصر} : (1, 3)$$

در اینجا، b_{11}, b_{12}, b_{13} مجهول هستند. ضمناً، توجه کنید که ماتریس ترانهاد

$$\mathbf{A}^T = \begin{matrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{matrix}$$

فرض کنید

$$\mathbf{B}_1 = \begin{matrix} b_{11} \\ b_{12} \\ b_{13} \end{matrix}$$

بنابراین،

$$\mathbf{A}^T \mathbf{B}_1 = \begin{matrix} 1 \\ 0 \\ 0 \end{matrix}$$

با روش حذف گوسی b_{11}, b_{12}, b_{13} را بدست می‌آوریم.
دومین سطر $\mathbf{B}\mathbf{A}$

$$b_{21}a_{11} + b_{22}a_{21} + b_{23}a_{31} = 0 \quad \text{عنصر} : (2, 1)$$

$$b_{21}a_{12} + b_{22}a_{22} + b_{23}a_{32} = 1 \quad \text{عنصر} : (2, 2)$$

$$b_{21}a_{13} + b_{22}a_{23} + b_{23}a_{33} = 0 \quad \text{عنصر} : (2, 3)$$

در اینجا، b_{21}, b_{22}, b_{23} مجهول هستند.

فرض کنید

$$\mathbf{B}_2 = \begin{matrix} b_{21} \\ b_{22} \\ b_{23} \end{matrix}$$

بنابراین،

$$\mathbf{A}^T \mathbf{B}_2 = \begin{matrix} 0 \\ 1 \\ 0 \end{matrix}$$

با روش حذف گوسی b_{21}, b_{22}, b_{23} را بدست می‌آوریم.
سومین سطر $\mathbf{B}\mathbf{A}$

$$b_{31}a_{11} + b_{32}a_{21} + b_{33}a_{31} = 0 \quad \text{عنصر} : (3, 1)$$

$$b_{31}a_{12} + b_{32}a_{22} + b_{33}a_{32} = 0 \quad \text{عنصر} : (3, 2)$$

$$b_{31}a_{13} + b_{32}a_{23} + b_{33}a_{33} = 1 \quad \text{عنصر} : (3, 3)$$

در اینجا، b_{31}, b_{32}, b_{33} مجهول هستند.

فرض کنید

$$\mathbf{B}_3 = \begin{matrix} b_{31} \\ b_{32} \\ b_{33} \end{matrix}$$

بنابراین،

$$\mathbf{A}^T \mathbf{B}_3 = \begin{matrix} 0 \\ 0 \\ 1 \end{matrix}$$

با روش حذف گوسی b_{31}, b_{32}, b_{33} را بدست می‌آوریم.
راه دیگر [۱] اضافه کردن ماتریس ضرایب با ماتریس همانی و سپس روش اعمال گوس-جردن بر ماتریس ضرایب ماتریس همانی است. ماتریس همانی اصلی می‌شود \mathbf{A}^{-1} . با ماتریس اضافه شده برای ماتریس ضرایب 3×3 بصورت زیر شروع می‌کنیم:

$$\left| \begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array} \right|$$

مثال ۶.۳

این روش با مثال زیر روشن می‌شود:

$$\begin{aligned} x_1 - 3x_2 + x_3 &= -4 \\ -3x_1 + 4x_2 + 3x_3 &= -10 \\ 2x_1 + 3x_2 - 2x_3 &= 18 \end{aligned}$$

ماتریس اضافه شده برابر است با

$$\mathbf{A}_{aug} = \left| \begin{array}{ccc|ccc} -1 & -3 & 1 & 1 & 0 & 0 \\ -3 & 4 & 3 & 0 & 1 & 0 \\ 2 & 3 & -2 & 0 & 0 & 1 \end{array} \right|$$

سطر اول را در $\frac{-3}{1}$ ضرب و از سطر دوم کم می‌کنیم، میدهد
 $(-3 + 3 \times 1), (4 - 3 \times 3), (3 + 3 \times 1), (0 + 3 \times 1), (1 + 3 \times 0), (0 + 3 \times 0) = (0, -5, 6, 3, 1, 0)$

سطر اول را در $\frac{2}{1} = 2$ ضرب و از سطر سوم کم می‌کنیم، میدهد
 $(2 - 2 \times 1), (3 - 2 \times (-3)), (-2 - 2 \times 1), (0 - 2 \times 1), (0 - 2 \times 0), (1 - 2 \times 0) = (0, 9, -4, -2, 1)$

$$\mathbf{A}_{aug, Equiv} = \left| \begin{array}{ccc|ccc} 1 & -3 & 1 & 1 & 0 & 0 \\ 0 & -5 & 6 & 3 & 1 & 0 \\ 0 & 9 & -4 & -2 & 0 & 1 \end{array} \right|$$

سطر دوم را در $\frac{-9}{5}$ و از سطر سوم کم می‌کنیم، میدهد
 $0 + \frac{9}{5} \times 0, 9 + \frac{9}{5} \times (-5), -4 + \frac{9}{5} \times 6, -2 + \frac{9}{5} \times 3, -4 + \frac{9}{5} \times 3, 0 + \frac{9}{5} \times 1, 1 + \frac{9}{5} \times 0$

$$= 0, 0, \frac{34}{5}, \frac{17}{5}, \frac{9}{5}, 1$$

$$\mathbf{A}_{aug, Equiv} = \left| \begin{array}{ccc|ccc} 1 & -3 & 1 & 1 & 0 & 0 \\ 0 & -5 & 6 & 3 & 1 & 0 \\ 0 & 0 & \frac{34}{5} & \frac{17}{5} & \frac{9}{5} & 1 \end{array} \right|$$

سطر سوم را در $\frac{30}{34}$ و از سطر دوم کم می‌کنیم، میدهد
 $0 - \frac{30}{34} \times 0, -5 - \frac{30}{34} \times 0, 6 - \frac{30}{34} \times \frac{34}{5}, 3 - \frac{30}{34} \times \frac{17}{5}, 1 - \frac{30}{34} \times \frac{9}{5},$

$$0 - \frac{30}{34} \times 1 = 0, -5, 0, 0, -\frac{20}{34}, \frac{30}{34}$$

سطر سوم را در $\frac{34}{5}$ و از سطر اول کم می‌کنیم، میدهد
 $1 - \frac{5}{34} \times 0, -3 - \frac{5}{34} \times 0, 1 - \frac{5}{34} \times \frac{34}{5}, 1 - \frac{5}{34} \times \frac{17}{5}, 0 - \frac{5}{34} \times \frac{9}{5},$

$$0 - \frac{5}{34} \times 1 = (1, -3, 0, \frac{1}{2}, -\frac{9}{34}, -\frac{5}{34})$$

$$\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|ccc} 1 & -3 & 0 & \frac{1}{2} & -\frac{9}{34} & -\frac{5}{34} \\ 0 & -5 & 0 & 0 & -\frac{20}{34} & -\frac{30}{34} \\ 0 & 0 & \frac{34}{5} & \frac{17}{5} & \frac{9}{5} & 1 \end{array}$$

سطر دوم را در $\frac{3}{5}$ و از سطر اول کم می‌کنیم، میدهد
 $1 - \frac{3}{5} \times 0, -3 - \frac{3}{5} \times (-5), 0 - \frac{3}{5} \times 0, \frac{1}{2} - \frac{3}{5} \times 0, -\frac{9}{34} + \frac{3}{5} \times \frac{20}{34},$

$$-\frac{5}{34} + \frac{3}{5} \times \frac{30}{34} = (1, 0, 0, \frac{1}{2}, \frac{3}{34}, \frac{13}{34})$$

$$\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{2} & \frac{3}{34} & \frac{13}{34} \\ 0 & -5 & 0 & 0 & -\frac{20}{34} & -\frac{30}{34} \\ 0 & 0 & \frac{34}{5} & \frac{17}{5} & \frac{9}{5} & 1 \end{array}$$

سطر دوم را بر -5 و سطر سوم بر $\frac{34}{5}$ تقسیم می‌کنیم، میدهد
 $\mathbf{A}_{aug, Equiv} = \begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{2} & \frac{3}{34} & \frac{13}{34} \\ 0 & 1 & 0 & 0 & \frac{4}{34} & \frac{6}{34} \\ 0 & 0 & 1 & \frac{1}{2} & \frac{9}{34} & \frac{5}{34} \end{array}$

بنابراین،

$$\mathbf{A}^{-1} = \begin{array}{ccc} \frac{1}{2} & \frac{3}{34} & \frac{13}{34} \\ 0 & \frac{4}{34} & \frac{6}{34} \\ \frac{1}{2} & \frac{9}{34} & \frac{5}{34} \end{array}$$

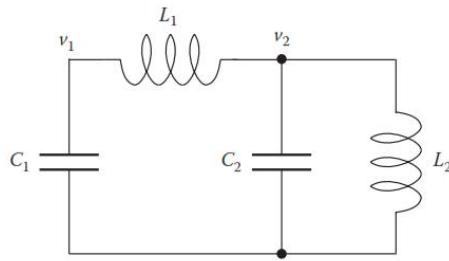
اثبات $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$ بعنوان تمرین بعهده دانشجویان قرار داده می‌شود.

۸.۳ مسئله مقدار ویژه

یکی از کاربردهای مسئله ویژه در مدارهای رزونانسی است. شبکه LC شکل (۳.۳) را در نظر گیرید. با استفاده از روابط مربوط به ولتاژ دوسر سیم پیچ $v = L \frac{di}{dt}$ و جریان درون خازن $i = C \frac{dv}{dt}$ معادله دیفرانسیلی حاکم بر مدار که توصیف کننده ولتاژ گره‌های v_1 و v_2 هستند، برابر است با

$$\frac{d^2v_1}{dt^2} = -\frac{1}{L_1C_1}v_1 + \frac{1}{L_1C_1}v_2 \quad (17.3)$$

$$\frac{d^2v_2}{dt^2} = \frac{1}{L_2C_2}v_1 - \frac{1}{L_2C_2}v_2 \quad (18.3)$$

شکل ۳.۳: مدار LC

میخواهیم مودهای نوسانی را بقسمی تعیین کنیم که هر دو ولتاژ با یک فرکانس نوسان کنند. برای بدست آوردن چنین جوابی، ولتاژها را بصورت زیر قرار میدهیم
 $v_1 = K_1 e^{j\omega t}$ (۱۹.۳)

$$v_2 = K_2 e^{j\omega t} \quad (۲۰.۳)$$

که در آن K_1 و K_2 ثابت‌هایی هستند که باید تعیین شوند. با قرار دادن رابطه ۱۹.۳ و ۲۰.۳ در معادلات ۱۸.۳ و ۱۷.۳ خواهیم داشت

$$\left(\frac{1}{L_1 C_1} - \omega^2 \right) K_1 - \frac{1}{L_1 C_1} K_2 = 0 \quad (۲۱.۳)$$

$$-\frac{1}{L_1 C_2} K_1 + \frac{1}{L_1 C_2} + \left(\frac{1}{L_2 C_2} - \omega^2 \right) K_2 = 0 \quad (۲۲.۳)$$

معادلات ۲۱.۳ و ۲۲.۳ دو معادله همگن جبر خطی با دو مجهول هستند. جبر خطی بما میگوید که تنها راه برای دو معادله جبر خطی همگن دو مجهولی که دارای جواب معقول دارند این است که دترمینان ضرایب ماتریس صفر شوند. بنابراین

$$\begin{vmatrix} \frac{1}{L_1 C_1} - \omega^2 & -\frac{1}{L_1 C_1} \\ -\frac{1}{L_1 C_2} & -\frac{1}{L_1 C_2} + \frac{1}{L_2 C_2} - \omega^2 \end{vmatrix} = 0 \quad (۲۳.۳)$$

اکنون فرض کنید $\omega^2 = \lambda$ و

$$\mathbf{A} = \begin{pmatrix} \frac{1}{L_1 C_1} & -\frac{1}{L_1 C_1} \\ -\frac{1}{L_1 C_2} & \frac{1}{L_1 C_2} + \frac{1}{L_2 C_2} \end{pmatrix}$$

$$\text{با بازنویسی رابطه ۲۳.۳ بر حسب } \lambda \text{ و عناصر } a_{ij} \text{ ماتریس } \mathbf{A} \text{ خواهیم داشت} \quad (۲۴.۳)$$

$$\lambda^2 - (a_{11} + a_{22})\lambda + (a_{11}a_{22} - a_{12}a_{21}) = 0$$

جواب معادله ۲۴.۳ مقادیر ویژه λ_1 و λ_2 ماتریس \mathbf{A} که مربع فرکانس‌های طبیعی این سیستم است را بما میدهد. نسبت دامنه نوسانات این دو ولتاژ را می‌توان با قرار دادن مقادیر λ در روابط ۲۱.۳ و ۲۲.۳ را بدست آورد.

$$\frac{K_2}{K_1} = -(a_{11} - \lambda_1)/a_{12} = 1 - L_1 C_1 \lambda_1$$

برای اولین مود، و

$$\frac{K_2}{K_1} = -(a_{11} - \lambda_2)/a_{12} = 1 - L_1 C_1 \lambda_2$$

برای دومین مود است.

بردار ویژه \mathbf{V}_1 وابسته به مقدار ویژه λ_1 برابر است با

$$\begin{aligned} K_1 \\ -K_1(a_{11} - \lambda_1)/a_{12} \end{aligned}$$

و بردار ویژه \mathbf{V}_2 وابسته به مقدار ویژه λ_2 برابر است با

$$\begin{aligned} K_1 \\ -K_1(a_{11} - \lambda_2)/a_{12} \end{aligned}$$

چون K_1 مقدار دلخواهی است می‌توان آنرا 1 در نظر گرفت، بنابراین

$$\mathbf{V}_1 = \begin{matrix} 1 \\ -(a_{11} - \lambda_1)/a_{12} \end{matrix}$$

۹

$$\mathbf{V}_2 = \begin{matrix} 1 \\ -(a_{11} - \lambda_2)/a_{12} \end{matrix}$$

اگر \mathbf{V} یک بردار ویژه ماتریس \mathbf{A} مربوط به مقادیر ویژه λ باشد، در این صورت $b\mathbf{V}$ نیز بردار ویژه محسوب می‌شود. در اینجا b مقدار ثابت غیر صفر است [?].

در متلب تابعی بنام `eig` دارد که مقادیر ویژه یک ماتریس مربعی را پیدا می‌کند. تعریف متلب برای این تابع چنین است:

$E = EIG(X)$ برداری است که شامل مقادیر ویژه ماتریس مربعی X است. $[V, D] = EIG(X)$ ماتریس قطری D مقادیر ویژه و ماتریس کامل V که هر ستونش بردار ویژه مربوطه را میدهد بطوریکه $X * V = V * D$

برای این مسئله، ماتریس A نمایشگر X است. بنابراین، اجرای $[V, D] = eig(A)$

بردارهای ویژه وابسته به مقادیر ویژه λ_1 و λ_2 را میدهد. $(V(:, 1)$ مربوط به λ_1 و $V(:, 2)$ مربوط به λ_2 است).

مثال ۷.۳:

فرض کنید در شکل (۳.۳) پارامترهای زیر داده شده‌اند:

$$L_1 = 15 \text{ mH}, \quad L_2 = 25 \text{ mH}, \quad C_1 = 1\mu\text{F}, \quad C_2 = 33\mu\text{F}$$

میخواهیم یک برنامه در متلب داشته باشیم که

۱. مقادیر ویژه دستگاه را با استفاده از معادلات ۲۱.۳ و ۲۲.۳ با استفاده از تابع `eig` بدست آورد.

۲. توابع ویژه با هر دو رابطه ۲۱.۳ و ۲۲.۳ و خروجی متلب $[V, D]$ بدست آورد.

برنامه چنین است.

```
% matrix_example.m
% Example_3_7.m: Eigenvalues and eigenvectors.
% L = eig(A) is a vector containing the eigenvalues of a
% square matrix A.
```

```
% [M,D] = eig(A) produces a diagonal matrix D of
% eigenvalues and a full matrix M whose columns are the
% corresponding eigenvectors such that A*L = M*D.
% Units are in SI units.
clear; clc;
L1=15e-3; L2=25e-3; % henries
C1=1e-6; C2=33e-6; % farads
A(1,1)=1/(L1*C1);
A(1,2)=-1/(L1*C1);
A(2,1)=-1/(L1*C2);
A(2,2)=1/(L1*C2) + 1/(L2*C2);
% Solve for eigenvalues and eigenvectors using derivation
% in Eqn 3.25:
% lambda^2-(A(1,1)+A(2,2))
% *lambda+(A(1,1)*A(2,2)-A(1,2)*A(2,1))=0
% Solve for lambda using quadratic formula:
b=-(A(1,1)+A(2,2)); c=A(1,1)*A(2,2)-A(1,2)*A(2,1);
lambda1=(-b+sqrt(b^2-4*c))/2;
lambda2=(-b-sqrt(b^2-4*c))/2;
% From Eqns 3.26 and 3.27:
eigenvector1=[1;-(A(1,1)-lambda1)/A(1,2)];
eigenvector2=[1;-(A(1,1)-lambda2)/A(1,2)];
fprintf('Eigenvalues solved via Equation 3.25: ');
fprintf('lambda1=% .0f lambda2=% .0f\n',lambda1,lambda2);
fprintf('Eigenvectors solved via Equation 3.26 ');
fprintf('and 3.27:\n');
eigenvector1
eigenvector2
% Solve for eigenvalues and eigenvectors using MATLAB's
% eig function. Calling eig and returning result to a
% scalar will give the eigenvalues.
L=eig(A);
fprintf('Eigenvalues via MATLAB eig function: ');
fprintf('L(1)=%.0f L(2)=%.0f\n',L(1),L(2));
% Calling eig and returning result to two variables will
% give the eigenvectors and a diagonal matrix of the
% eigenvalues.
[V D] = eig(A);
fprintf('Eigenvectors via MATLAB eig function:\n');
V1=V(:,1)
V2=V(:,2)
D
%-----
```

نتایج زیر توسط متلب تولید شده است:

```

Eigenvalues solved via Equation 3.25: lambda1 = 68723140
lambda2 = 1175850
Eigenvectors solved via Equation 3.26 and 3.27:
eigenvector1 =
1.0000
-0.0308
eigenvector2 =
1.0000
0.9824
Eigenvalues via MATLAB eig function: L(1) = 68723140 L(2) =
1175850
Eigenvectors via MATLAB eig function:
V1 =
0.9995
-0.0308
V2 =
0.7134
0.7008
D =
1.0e+007 *
6.8723 0
0 0.1176
%-----

```

با آزمونی جواب، مشاهده می‌شود که $\lambda_2 = D_{(2,2)}$ و $\lambda_1 = D_{(1,1)}$ است. همچنین، ضریب اسکالری از V_1 و $eigenvector2$ ضریب اسکالری از V_2 است.

مسائل

۱. با استفاده از روش حذف گوسی دستگاه‌های زیر را حل کنید:

الف •

$$\begin{aligned} 2x_1 + 3x_2 - x_3 &= 20 \\ 4x_1 - x_2 + 3x_3 &= -14 \\ x_1 + 5x_2 + x_3 &= 21 \end{aligned}$$

ب •

$$\begin{aligned} 4x_1 + 8x_2 + x_3 &= 8 \\ -2x_1 - 3x_2 + 2x_3 &= 14 \\ x_1 + 3x_2 + 4x_3 &= 30 \end{aligned}$$

ج •

$$\begin{aligned} 2x_1 + 3x_2 + x_3 - x_4 &= 1 \\ 5x_1 - 2x_2 + 5x_3 - 4x_4 &= 5 \\ x_1 - 2x_2 + 3x_3 - 3x_4 &= 3 \\ 3x_1 + 8x_2 - x_3 + x_4 &= -1 \end{aligned}$$

پروژه‌ها

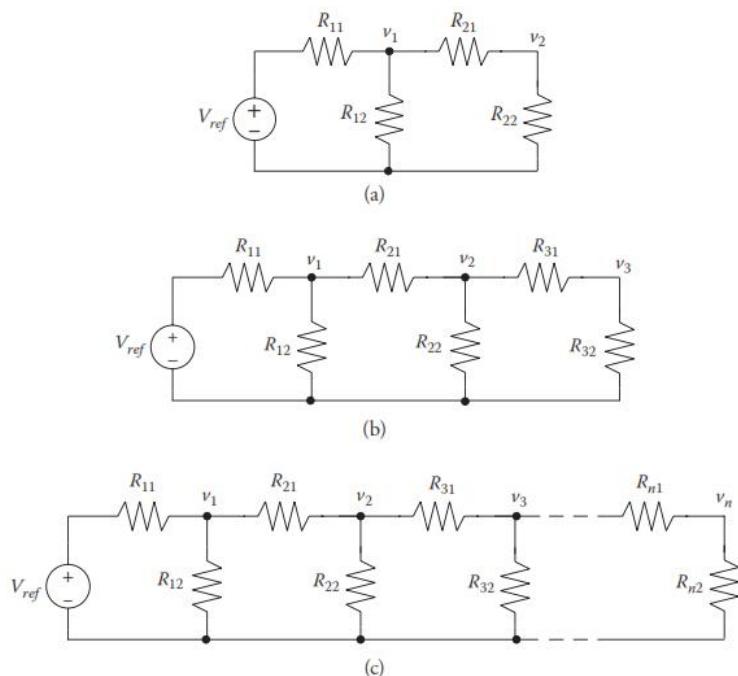
۱.۳ پروژه

برای مدار شکل (۱.۳)، پنج جریان مجهول را بجای سه ولتاژ مجهول حل کنید. معادلات جبری را با استفاده از روابط ۴.۳، ۲.۳، و ۵.۳ و با استفاده از قانون ولتاژهای کیرشef (KVL) در نوشتن دو معادله اضافی بدست آورید. قانون KVL میگوید که جمع افت ولتاژها در هر حلقه در مدار باید صفر باشد. معادلات KVL را برای دو حلقه در مدار $V_1 \rightarrow R_1 \rightarrow R_2 \rightarrow R_4 \rightarrow R_5 \rightarrow V_2$ و $R_3 \rightarrow R_4 \rightarrow R_2$ بنویسید. شما باید در آخر پنج معادله و پنج مجهول داشته باشید. با استفاده از روش ماتریس معکوس متلب این را حل کرده و جوابهای خود را با نتایج مثال ۳.۳ تائید کنید.

۲.۳ پروژه

شکل (۴.۳) یک مدار مقاومتی بنام "شبکه‌نربانی" را نشان می‌دهد.

- با استفاده از قانون جریانهای کیرشef، یک دستگاه معادلات برای شبکه مرتبه دوم و سوم شکل‌های (۴.۳)(الف) و (۴.۳)(ب) بنویسید. بعلاوه، معادلاتی نیز برای شبکه مرتبه چهارم بنویسید.



شکل ۴.۳: (الف) شبکه نربانی مرتبه دوم. (ب) شبکه نربانی مرتبه سوم. (ج) شبکه نربانی مرتبه n ام.

- در بخش اول باید الگوی یکسان شدن نتایج را مشاهده کرده باشید. با استفاده از این الگوی، ماتریسی برای شبکه نربانی مرتبه هشت بنویسید. ولتاژ برای همه مدارها را با

استفاده از مطلب $V_{ref} = 5V$ و مقادیر مقاومت‌ها را بقرار زیر در نظر گیرید:

$$\begin{aligned} R_{11} &= 2200, & R_{12} &= 2200 \\ R_{21} &= 1200, & R_{22} &= 6800 \\ R_{31} &= 3900, & R_{32} &= 2200 \\ R_{41} &= 3300, & R_{42} &= 5700 \\ R_{51} &= 1800, & R_{52} &= 5100 \\ R_{61} &= 4700, & R_{62} &= 3900 \\ R_{71} &= 5700, & R_{72} &= 6800 \\ R_{81} &= 1200, & R_{82} &= 2700 \end{aligned}$$

۳. برای حل این مسئله، یک برنامه بزبان متلب نوشته بطوری که بصورت خودکار برای هر مقدار دلخواه n و R_0 (مانند شکل ۴.۲) تحت شرایط زیر آنرا حل نماید:

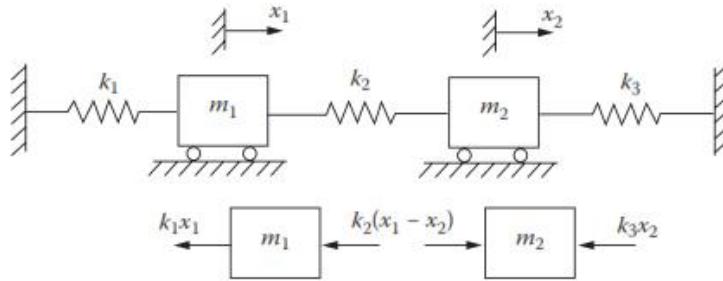
$$\begin{aligned} R_0 &= R_{11} = R_{21} = R_{31} = \dots = R_{(n-1)1} = R_{n1} \\ 2R_0 &= R_{12} = R_{22} = R_{32} = \dots = R_{(n-1)2} \end{aligned}$$

جوابها را برای دو حالت زیر بدست آورید:

$$\begin{aligned} n &= 16, & R_0 &= 1000\Omega, & V_{ref} &= 10 \\ n &= 24, & R_0 &= 100\Omega, & V_{ref} &= 1 \end{aligned}$$

۳.۳ پروژه

یکی از مسائل مهم مسائل مقادیر ویژه، تئوری نوسان مکانیکی است. سیستمی با دو جرم (m_1) و سه فنر (با ثابت‌های k_1 , k_2 , و k_3) مطابق شکل ۵.۳ در نظر گیرید.



شکل ۵.۳: مسئله نوسان مکانیکی با دو درجه آزادی.

معادلات دیفرانسیلی حاکم که توصیف کننده حرکت دو جرم است عبارتند از

$$m_1 \ddot{x}_1 = k_2(x_2 - x_1) - k_1 x_1 \quad (۵.۳)$$

$$m_2 \ddot{x}_2 = -k_2(x_2 - x_1) - k_3 x_2 \quad (۵.۴)$$

با استفاده از تابع eig مطلب، مودهای نوسانی اینگونه حرکت هماهنگ (هارمونیک) را در فرکانس یکسان تعیین کنید. مقادیر را بقرار زیر در نظر گیرید:

$$m_1 = m_2 = 1500 \text{ kg}, \quad k_1 = 3250 \text{ N/m}, \quad k_2 = 3500 \text{ N/m}, \quad k_3 = 3000 \text{ N/m}$$

۴.۳ پروژه

فرض کنید شخص تولیدکننده‌ای میخواهد یک دستگاه را به قیمت 40,000 دolar خریداری کند. وی قصد دارد پول را از بانک قرض کرده و قسطهای آنرا ده ساله در 120 قسط مساوی پرداخت

کند. نرخ بهره سالیانه شش درصد است. هرماه، سود به روی قسط وام حساب می‌شود. او می‌خواهد بداند هر ماہ باید چه قسطی را پرداخت نماید. این مسئله را می‌توانید با استفاده از دستگاه معادلات خطی حل کنید.

فرض کنید x_j مقدار قسط زام است که برای پرداخت اصل پول است. در این صورت، معادله توصیف قسط برابر است با

$$M = x_1 + P - \sum_{n=1}^{n=j-1} x_n I \quad (26.3)$$

که در آن

$$\begin{aligned} M &= \text{قسط ماهانه} \\ P &= \text{مبلغ قرض گرفته شده} \\ I &= \text{نرخ سود ماهانه} \end{aligned}$$

تعداد کل مجھولات ۱۲۱ (مقدار $120x_j$ و M). با اعمال معادله 26.3 به هر ماه ۱۲۰ معادله میدهد. ۱۲۱ کمین معادله برابر است با

$$P = \sum_{n=1}^{n=120} x_n \quad (27.3)$$

برنامه‌ای بزیان مطلب نوشته که:

۱. از کاربر سوال کند که از طریق صفحه کلید، مقدار وام P ، نرخ سود سالیانه I ، و دوره زمانی Y (بر حسب سال) را وارد کند.

۲. با استفاده از ماتریس ضرایب $A_{n,m}$ دستگاه معادلات خطی، دستگاه معادلات خطی را تدوین کنید. n نمایشگر تعداد معادلات، و m نمایشگر ضرایب x_m در معادلات است. $x_{121} = M$ در نظر گیرید.

۳. دستگاه معادلات خطی را در مطلب حل کنید.

۴. جدولی شامل چهار ستون چاپ کنید. اولین ستون باید تعداد ماهها، دومین ستون قسط ماهانه، سومین ستون مقدار قسطی که از بابت مبلغ اصلی پرداخت می‌شود، و چهارمین ستون مقدار سود قسط برای آن ماه باشد.

۵.۳ پروژه

فرامین `tic` و `toc` در مطلب بشما اجازه میدهد که زمان اجرای برنامه خود را برای یک سری فرامین اندازه‌گیری کنید. `tic` و `toc` بعنوان کرونومتر (قدم شمار) رفتار می‌کنند؛ `tic` زمان شروع و `toc` خاتمه است که گذر زمان را چاپ می‌کند. مثلاً

```
tic;
a = factorial(25);
toc;
```

اجرای این متن! ۲۵ را محاسبه کرده و سپس زمان کل اجرای برنامه را چاپ می‌نماید:

```
Elapsed time is 0.005509 seconds.
```

البته، زمانی که شما مشاهده می‌کنید بستگی به سرعت کامپیوتر شما دارد.

۱. برنامه‌ای بزیان متلب نوشته که ماتریس A تصادفی $n \times n$ که هر عنصر یک عدد صحیح تصادفی بین $-E_{max}$ و E_{max} است داشته باشد. با استفاده از توابع `rand` و `round` متلب این را انجام دهید. برای آموزش استفاده از این توابع در پنجره فرمان متلب `help rand` و `help round` را تایپ کنید. فرض کنید در ابتدا $n = 100$ و $E_{max} = 100$ است.

۲. ضمناً، بردار C با عناصر تصادفی ستونی $1 \times n$ با استفاده از E_{max} و n بالا تولید کنید.

۳. معادله $AX = C$ را بدو طریق حل کنید: (الف) با پیدا کردن A^{-1} (با تابع `inv`) و ضرب در بردار C و (ب) با استفاده از حذف گوسی (عملگر ' \backslash ' متلب). از `toc` و `tic` برای اندازه‌گیری زمان اجرای هر یک از روش‌های فوق استفاده کنید. آیا تفاوت جواب‌ها را در کمی کنید؟

۴. این برنامه را ده بار اجرا کرده و زمان متوسط را برای هر دو روش بدست آورید. همچنین زمان متوسط اجرا را بازاء $n = 3, 10, 30, 300, 1000$ بدست آورید. زمان اجرا را در مقابل محور n در مقیاس $\log - \log$ رسم کنید.

فصل ۴

ریشه‌های جبری و معادلات متعالی

۱.۴ مقدمه

در تحلیل مسائل مهندسی مختلف، ما اغلب مجبور می‌شویم که ریشه‌های معادلاتی که حل تحلیلی براحتی بدست نمی‌آید را بدست آوریم. برای مثال ریشه معادلات چند جمله‌ای درجه n و معادلات متعالی^۱ که شامل توابع مثلثاتی، نمائی، یا لگاریتمی است. در این فصل، چندین روش را برای حل عددی اینگونه معادلات همراه با دقت و پیچیدگی ریاضی ارائه می‌کنیم. هم‌چنین، یک بخش را به تابع `fzero` متلب اختصاص داده و می‌توان با استفاده از این تابع ریشه‌های معادلاتی که قبلًاً اشاره شد را بدست آورد.

۲.۴ روش جستجو

معادله‌ای که ریشه‌های آن را باید بدست آورد باید آنرا بصورت استاندارد زیر نشان داد:

$$f(x) = 0 \quad (1.4)$$

بصورت زیر اقدام می‌کنیم: اول، جستجوئی انجام داده تا ببینیم ریشه‌های حقیقی در چه محدوده‌ای قرار دارند. این را با تقسیم کردن ناحیه x به N زیر قطعه مساوی شروع می‌کنیم، خواهیم داشت

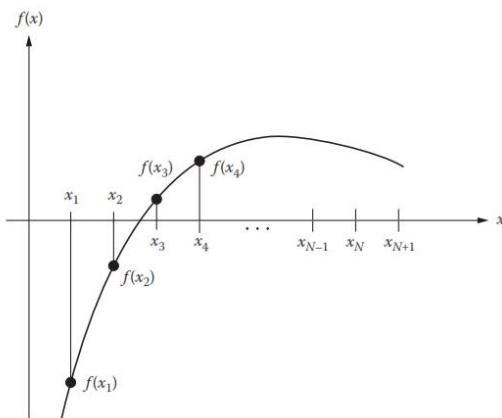
$$x_1, x_2, x_3, \dots, x_{N+1} \quad \text{و} \quad x_{i+1} = x_i + \Delta x$$

در این صورت، جائی که (x) f تغییر علامت میدهد (به شکل ۱.۴ مراجعه کنید) معلوم کنید. این وقتی اتفاق می‌افتد که علامت دو مقادیر متوالی $f(x)$ مخالف شوند، یعنی،

$$f(x_i)f(x_{i+1}) < 0$$

تغییر علامت معمولاً نشاندهنده این است که ریشه حقیقی گذر کرده است. البته، ممکن است نشانگر ناپیوستگی در تابع باشد. (مثال: $\tan x = \pi/2$ در $x = \pi/2$ ناپیوسته است.) وقتی فوایدی که ریشه‌ها در آنجا قرار دارند معلوم شد، چندین روش مختلف برای تعیین ریشه‌های حقیقی وجود دارد.

^۱transcendental equations



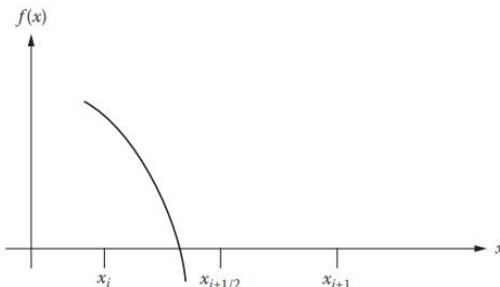
شکل ۱.۴: انتخاب فواصلی که ریشه در آنجا قرار دارد. در این حالت $f(x_2)f(x_3) < 0$, و لذا ریشه بین x_2 و x_3 قرار دارد.

۳.۴ روش دوبخشی

فرض کنید یک ریشه بین x_i و x_{i+1} قرار دارد. در این صورت، این فاصله را بدو قسمت تقسیم میکنیم (به شکل ۲.۴ مراجعه کنید)، و بنابراین

$$x_{i+1/2} = x_i + \frac{\Delta x}{2}$$

اکنون $f(x_i)f(x_{i+1/2})$ را محاسبه میکنیم:



شکل ۲.۴: انتخاب محدوده‌ای که ریشه در روش دوبخشی قرار دارد.

- حالت اول: اگر $f(x_i)f(x_{i+1/2}) < 0$ باشد، در این صورت ریشه بین x_i و $x_{i+1/2}$ قرار دارد.
- حالت دوم: اگر $f(x_i)f(x_{i+1/2}) > 0$ باشد، در این صورت ریشه بین $x_{i+1/2}$ و x_{i+1} قرار دارد.
- حالت سوم: اگر $f(x_i)f(x_{i+1/2}) = 0$ باشد، در این صورت $x_{i+1/2}$ یا x_i ریشه حقیقی است.

برای حالت‌های اول و دوم، بازه شامل ریشه را انتخاب کرده و فرآیند را تکرار میکنیم. برای تکرار فرآیند، مثلاً r بار، در این صورت $\Delta x_f = \frac{\Delta x}{2^r}$ ، که در آن Δx اندازه بازه اولیه است که قبل از فرآیند دوبخشی^۲، ریشه در آن بازه قرار دارد و Δx_f اندازه بازه بعداز r دوبخشی است. اگر

²bisection process

$(\Delta x)_f$) بقدر کافی کوچک باشد، در این صورت تقریب خیلی خوبی برای ریشه است که جائی در بین آخرین فاصله دوبخشی مثلاً وسط فاصله قرار دارد.

مثال: برای بیست بار دوبخشی،

$$(\Delta x)_f = \frac{\Delta x}{2^{20}} \approx \Delta x \times 10^{-6}$$

است. روش برنامه بقرار زیر است:

فرض کنید

$$x_A = x_i$$

$$x_B = x_{i+1}$$

$$x_C = \frac{1}{2}(x_A + x_B)$$

اگر $f(x_A)f(x_C) < 0$ (نشاندهنده این است که ریشه بین x_A و x_C است)، در این صورت، فرض کنید

$$x_B = x_C$$

$$x_C = \frac{1}{2}(x_A + x_B)$$

و فرایند را تکرار کنید.

در غیر این صورت، اگر $f(x_A)f(x_C) > 0$ باشد (نشاندهنده این است که ریشه بین x_B و x_C قرار دارد)،

در این صورت، فرض کنید

$$x_A = x_C$$

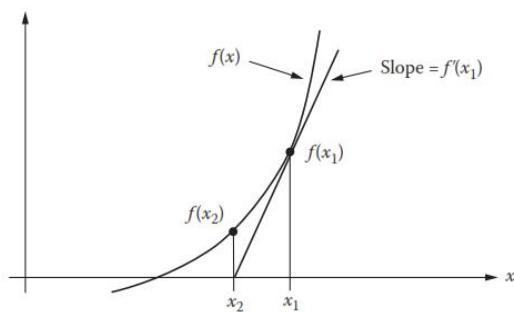
$$x_C = \frac{1}{2}(x_A + x_B)$$

و فرایند را تکرار کنید.

اگر $f(x_A)f(x_C) = 0$ باشد، در این صورت x_A یا x_C ریشه است.

۴.۴ روش نیوتون-رافسون

روش روش نیوتون-رافسون^۳ از مماس بر منحنی $f(x) = 0$ برای تخمین ریشه استفاده می‌کند. لازم است رابطه‌ای برای $f(x)$ و حدس اولیه برای ریشه، مثلاً x_1 (به شکل ۳.۴ مراجعه کنید). شیب مماس بر منحنی را در نقطه x_1 میدهد.



شکل ۳.۴: پیشگوئی ریشه در روش نیوتون-رافسون.

مماس بر منحنی داریم

$$\frac{f(x_1) - f(x_2)}{x_1 - x_2} = f'(x_1) \quad (2.4)$$

^۳Newton-Raphson Method

حدس اولیه ما (یا تکرار) برای تعیین ریشه با قرار دادن $0 = f(x_2)$ و حل معادله برای x_2 است، داریم،

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \quad (3.4)$$

آزمایش کنید که آیا $\epsilon < |f(x_2)|$ هست. در اینجا ϵ تولرانس خطأ است.
اگر جواب صحیح است، در این صورت x_2 ریشه است و لذا x_2 چاپ شود.
اگر جواب صحیح نیست، $x_1 = x_2$ قرار داده و فرایند تکرار شود.
این فرایند آنقدر تکرار شود تا $\epsilon < |f(x_n)|$ بdest است. n تعداد تکرارها است.
راه دیگر همگرائی چنین است:

$$\left| \frac{f(x_1)}{f'(x_1)} \right| < \epsilon \quad (4.4)$$

معمولًا، ϵ باید بقدر کافی، مثلاً 10^{-4} ، کوچک باشد. اما بزرگ یا کوچک بودن این عدد بستگی به این دارد که شما چقدر میخواهید به ریشه حقیقی نزدیک باشید. شما میتوانید دقیق ریشه تعیین شده را با قرار دادن آن درتابع $f(x)$ مشاهده نمایید. و هر قدر این مقدار به صفر نزدیکتر باشد، بهتر است.
روش نیوتون-رافسون برای همگرائی سریع بکار برد میشود. البته، حالت‌های وجود دارد که همگرائی حاصل نمی‌شود. این وقتی اتفاق می‌افتد که

- الف: $f'(x)$ نزدیک ریشه تغییر علامت دهد.

- ب- حدس اولیه برای ریشه حقیقی دور از ریشه حقیقی باشد.

اگر روش نیوتون-رافسون را با روش جستجو برای تعیین فواصلی که ریشه حقیقی در آنجا قرار دارد ترکیب کنیم، همگرائی مسئله‌ای نخواهد بود.

۵.۴ توابع roots و fzero در متلب

متلب دارای توابعی است که ریشه‌های حقیقی یکتابع یک متغیره، از قبیل توابع متعالی، یا چندجمله‌ای درجه n را تعیین می‌کند. تابع `fzero` برای معادلات متعالی، و تابع `roots` برای معادلات چندجمله‌ای بکار برد می‌شود. ابتداء تابع `fzero` را بررسی می‌کنیم.

۱.۵.۴ تابع fzero

برای شروع، در پنجره فرمان متلب عبارت زیر را تایپ کنید

```
>> help fzero
```

این چندین توصیف مختلف را بشما میدهد، در اینجا دو تای اول آورده می‌شود

`FZERO` Single-variable nonlinear zero finding.

```
X = FZERO(FUN,X0) tries to find a zero of the function FUN
near X0, if X0 is a scalar. It first finds an interval
containing X0 where the function values of the interval
endpoints differ in sign, then searches that interval for
a zero. FUN is a function handle. FUN accepts real scalar
input X and returns a real scalar function value F,
evaluated at X. The value X returned by FZERO is near a
point where FUN changes sign (if FUN is continuous), or
```

`Nan if the search fails.`

`X = FZERO(FUN,X0), where X0 is a vector of length 2,
assumes X0 is a finite interval where the sign of
FUN(X0(1)) differs from the sign of FUN(X0(2)). An error
occurs if this is not true. Calling FZERO with a finite
interval guarantees FZERO will return a value near a point
where FUN changes sign.`

بطوری که گفته شد، `fzero` چندین استفاده مختلف دارد که بستگی به حدس اولیه X_0 ، (یک عدد اسکالر است) یا اگر شما بازه برای صفر (یعنی X_0 یک بردار با طول 2 است) را میدانید دارد. اولین استفاده مناسب است اگر شما اطلاعاتی درباره بازه ریشه دارید. استفاده دوم وقتی مناسب است که بیش از یک ریشه وجود دارد، و تمام ریشه‌ها را شما نیاز دارید. استفاده دوم را می‌توان با روش جستجو که در بخش قبل بحث شد ترکیب کرد.
استفاده نخست از تابع `fzero` چنین است

```
X = FZERO(FUN,X0)
```

تابعی است که می‌بایست آنرا حل کرده و میتواند نام فایل تابع (با پسوند `.m`) یا تابع بی‌نام (مانند آنچه در بخش ۸.۲ تشریح شد) باشد. X_0 حدس اولیه برای ریشه است و X جواب محاسبه شده بوسیله متلب است. بنابراین، فرض کنید نام فایل تابعی که میخواهیم ریشه‌اش را تعیین کنیم `myfun.m` و حدس اولیه ما برای ریشه ۳.۰ باشد. در این صورت، میتوان نوشت

```
X = fzero('myfun',3.0)
```

راه دیگر و معادل برای فرمان چنین است

```
X = fzero(@myfun,3.0)
```

هیچ علامت اضافی دیگری لازم ندارد.
اگر هیچ ریشه‌ای پیدا نشود، `fzero` جواب `NaN` (جواب ندارد) را میدهد.
استفاده دوم از `fzero` یک بردار بطول 2 است که در بازه‌ای که میخواهیم `FUN` علامتش را تغییر دهد، یعنی مقدار تابع `FUN` در $(X_0(1), X_0(2))$ علامتش مخالف مقدار مقدار تابع بازه $(X_0(1), X_0(2))$ است. اگر این شرط حاصل نشود، متلب خطای میدهد.

در برخی مواقع، میخواهیم صفر تابعی در دو آرگومان، مثلاً X و P که در آن P یک پارامتر و ثابت است بدست آوریم. برای حل با `fzero` P باید در برنامه تعریف شود. مثلاً، فرض کنید `myfun` در ام فایل متلب بصورت تابع زیر تعریف شده است:

```
function f = myfun(X,P)
f = cos(P*X);
```

عبارة `fzero` برای این منظور چنین بکار می‌رود

```
P = 1000;
root = fzero(@(X) myfun(X,P),X0)
```

که در آن `root` صفر تابع `myfun` است وقتی $P = 1000$ باشد. توجه کنید که P باید قبل از آنکه تابع `fzero` خوانده شود، تعریف گردد.

راه دیگر اضافه کردن پارامتر P به آرگومان تابع $myfun$ استفاده از بیان `global` در متلب است که پارامتر P را در تابع $myfun$ خارج از تابع تعریف نماید. عبارت `global` باید در هر دو تابع $myfun$ و برنامه اصلی بکار رود.
مثالاً:

در برنامه اصلی:

```
global P;
P = 1000;
X0 = 10.0;
root = fzero(@myfun,X0);
%-----
```

: *myfun.m* و در تابع

```
function f = myfun(x)
global P;
f = cos(P*x);
%-----
```

مثالی از دو روش بقرار زیر است.

مثال ۱.۴: استفاده از تابع `fzero` برای پیدا کردن اولین ریشه یک تابع متعالی

در بخش ۱۷.۲ در یک مدار RLC معادله حاکم برای ولتاژ $v(t)$ بدست آورده شد. برای حالتی که

$$v(t) = \exp\left(-\frac{1}{2RC}t\right) \left[A \cos\left(\sqrt{\frac{1}{LC} - \left(\frac{1}{2RC}\right)^2}t\right) + B \sin\left(\sqrt{\frac{1}{LC} - \left(\frac{1}{2RC}\right)^2}t\right) \right] \quad (۵.۴)$$

کمترین زمانی که $v(t)$ صفر است، یعنی پیدا کردن اولین گذر صفر ولتاژ $v(t)$ را در روی محور t را تعیین نمائید. پارامترهای زیر را فرض کنید:

$$R = 100 \Omega, \quad L = 10^{-3} H, \quad C = 10^{-6} F, \quad A = 6.0 V, \quad B = -9.0 V, \quad 0 \leq t \leq 0.5 \text{ msec}$$

```
% Example_4_1.m
% This script determines the initial zero crossing of the
% underdamped response of a parallel RLC circuit. The
% underdamped response is:
% v(t)=exp(-t/(2RC))*(A*cos(sqrt(1/LC-(1/2RC)^2)t)
% + B*sin(sqrt(1/LC-(1/2RC)^2)t))
% The circuit component values and initial conditions
% are:
% R=100 ohms; L=1.0e-3 henry; C=1.0e-6 farads;
% A= 6.0 volt,B=-9.0 volt
% Solve over the interval 0 <= t <= 0.5e-3 sec
clear; clc;
R=100; L=1.0e-3; C=1.0e-6; A=6.0; B=-9;
fprintf('Example 4.1: Find first zero crossing of ');
fprintf('underdamped RLC circuit\n');
```

```
tmin=0.0; tmax=0.5e-3;
% split up timespan into 100 intervals:
N=100;
dt=(tmax-tmin)/N;
% First, calculate t and v(t) at each timestep
for n=1:N+1
t(n) = tmin+(n-1)*dt;
v(n) = func_RLC(t(n),A,B,R,L,C);
fprintf('%10.4e %10.3f \n',t(n),v(n));
end
plot(t,v), xlabel('t'), ylabel('v'),
title('v vs t for a RLC circuit'), grid;
% Next, use the search method to find the first timestep
% where the sign of v(t) changes. When found, use fzero
% to solve.
for n=1:N
sign = v(n)*v(n+1);
if sign <= 0.0
root_interval(1)=t(n);
root_interval(2)=t(n+1);
% Solve for the zero of the multi-argument
% function func_RLC.m by invoking func_RLC as an
% anonymous function of one variable.
root = fzero(@(t) func_RLC(t,A,B,R,L,C),...
root_interval);
break;
end
end
fprintf('\n\n');
if n < N
fprintf('First zero crossing is at t=%10.4e s\n',...
root);
else
fprintf('No roots lie within %g <= t <= %g s\n',...
tmin,tmax);
end
% Check solution. Putting root into func_RLC should
% produce zero.
v = func_RLC(root,A,B,R,L,C);
fprintf('v=%8.6e \n',v);
-----
% func_RLC.m
% This function works with Example_4_1.m
function v=func_RLC(t,A,B,R,L,C)
arg1=1/(2*R*C);
```

```

arg3=sqrt(1/(L*C)-1/(2*R*C)^2);
% Equation (4.5):
v=exp(-arg1*t)*(A*cos(arg3*t)+B*sin(arg3*t));
%-----

```

نتایج برنامه

First zero crossing is at t = 1.8831e-005 s

میتوانیم مثال ۱.۴ را تغییر و اصلاح و از عبارت `global` برای پارامترها در تابع استفاده کنیم. برای این منظور،

۱. درست بعذار عبارتهای `global` و `clc` عبارت `global` را در برنامه اصلی و تابع اضافه کنید.
۲. تابع را با عبارت زیر

```
function v = func_RLC(t,A,B,R,L,C)
```

بجای

```
function v = func_RLC(t)
```

عوض کنید.

۳. در برنامه اصلی عبارت

```
v(n) = func_RLC(t(n),A,B,R,L,C);
```

بجای

```
v(n) = func_RLC(t(n));
```

عوض کنید.

۴. در برنامه اصلی عبارت

```
root = fzero(@(t) func_RLC(t,A,B,R,L,C),root_inteval);
```

بجای

```
root = fzero('func_RLC',root_inteval);
```

۲.۵.۴ تابع roots

عرض کنید. اکنون برنامه را اجرا کرده تا مشاهده کنید همان جواب را میدهد.

در متلب تابعی برای تعیین ریشه‌های چندجمله‌ای‌ها دارد. نام آن تابع `roots` است. برای بدست آوردن نحوه استفاده از تابع، در پنجره فرمان متلب عبارت زیر را بنویسید

```
>> help fzero
```

این برشما میدهد

`ROOTS` Find polynomial roots.

`ROOTS(C)` computes the roots of the polynomial whose coefficients are the elements of the vector `C`. If `C` has $N+1$ components, the polynomial is $C(1)*X^N + \dots + C(N)*X + C(N+1)$.

بنابراین، برای پیدا کردن ریشه‌های چندجمله‌ای $Ax^4 + Bx^3 + Cx^2 + Dx + E = 0$ ، عبارت `roots([A B C D E])` را اجرا می‌کنیم. تابع `roots` ریشه‌های حقیقی و موهومی چندجمله‌ای را بما میدهد.

برخی از توابع مفید در متلب عبارتند از

| | |
|----------------------|--|
| <code>poly(V)</code> | ضرایب چند جمله‌ای که ریشه‌های آن <code>V</code> است را میدهد |
| <code>real(X)</code> | بخش حقیقی را میدهد |
| <code>imag(X)</code> | بخش موهومی را میدهد |

مثال ۴.۴: استفاده از تابع `roots` برای پیدا کردن ریشه‌های یک چندجمله‌ای

```
% Example_4_2.m
% This program determines the roots of a polynomial using
% the built in function 'roots'.
% The first polynomial is: f=x^3-4.7*x^2-35.1*x+85.176.
% The roots of this polynomial are all real.
% The second polynomial is: f=x^3-9*x^2+23*x-65. The
% roots of this polynomial are both real and complex.
% Complex roots must be complex conjugates.
% To obtain more info on complex numbers in MATLAB, run
% "help complex" in the Command Window.
clear; clc;
% Define coefficients of first polynomial (real roots)
C(1)=1.0; C(2)=-4.7; C(3)=-35.1; C(4)=85.176;
fprintf('The first polynomial coefficients are:\n');
C
fprintf('The roots are: \n');
V=roots(C)
fprintf('The recalculated coefficients of the ');
fprintf('polynomial whose roots are V are:\n');
C_recalc=poly(V)
fprintf('\n\n');
% Define coefficients of second polynomial (real and
```

```
% complex roots)
D(1)=1.0; D(2)=-9.0; D(3)=23.0; D(4)=-65.0;
fprintf('The second polynomial coefficients are:\n');
D
fprintf('The roots are: \n');
W=roots(D)
fprintf('The real and imaginary parts of the ');
fprintf('roots are:\n');
re=real(W)
im=imag(W)
fprintf('The recalculated coefficients of the ');
fprintf('polynomial whose roots are W are:\n');
W_recalc=poly(W)
```

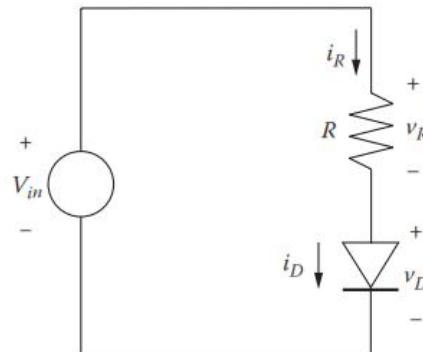
پروژه‌ها

۱.۴ پروژه

رابطه ولتاژ-جریان یک دیود نیمه‌هادی PN را می‌توان بصورت زیر نوشت:

$$i_D = I_S e^{\frac{q}{kT} v_D} - 1 \quad (6.4)$$

که در آن i_D و v_D همانطوری که در شکل (۴.۴) نشان داده شده است، بترتیب جریان و ولتاژ



شکل ۴.۴: مدار دیود- مقاومت

دیود، I_S ثابت و واحد آن آمپر است، که از ناخالصی نیمه‌هادی بدست می‌آید؛ $q = 1.6 \times 10^{-19}$ بار الکتریکی بر حسب کولمب است؛ $k = 1.38 \times 10^{-23}$ جول بر درجه کلوین میباشد؛ و T دمای مطلق بر حسب کلوین است.

می‌خواهیم جریان و ولتاژ درون این مدار را تعیین کنیم. با استفاده از قانون ولتاژهای کیرشوف، میدانیم که جمع جبری ولتاژها در مدار باید صفر باشد.

$$i_D = I_S e^{\frac{q}{kT} v_D} - 1 \quad (7.4)$$

با اعمال قانون اهم برای مقاومت، $v_R = i_R R$ و با توجه این که جریان مقاومت برابر جریان دیود است، میتوانیم رابطه ۷.۴ را بصورت زیر بنویسیم:

$$V_{in} - I_S (e^{\frac{q}{kT} v_D} - 1)R - v_D = 0 \quad (8.4)$$

فرض کنید

$$f(v_D) = V_{in} - I_S(e^{\frac{q}{kT}v_D} - 1)R - v_D$$

۱. تابعی برای $f(v_D)$ بزبان متلب نوشته و آنرا برای بازه v_D از صفر تا ۰.۸ ولت در قدمهای ده میلی ولت (۸۰ قسمت برای محدوده v_D) رسم کنید.

۲. با استفاده از روش جستجو، ناحیه کوچکی را برای معادله (۸.۴) بدست آورید که ریشه در آنجا قرار دارد.

۳. با استفاده از تابع `fzero` با دقت بیشتر ریشه را بدست آورید.
پارامترهای زیر را در این پروژه استفاده کنید:

$$T = 300 \text{ K}, \quad I_S = 10^{-14} \text{ A}, \quad V_{in} = 5 \text{ V}, \quad R = 1000 \Omega.$$

۲.۴ پروژه

میخواهیم مشخصه انتقالی DC را برای یک دیود شکل (۴.۴) تعیین کنیم. ولتاژ V_{in} را بعنوان پارامتر در نظر می‌گیریم. در اینجا V_{in} از پنج ولت تا دوازده ولت در قدمهای یک ولتی تغییر می‌کند. میخواهیم مقدار v_D برای تمام مقادیر V_{in} بدست آوریم.

برنامه‌ای بزبان متلب نوشته که ریشه‌های $f(v_D) = 0$ را تعیین نماید. در اینجا

$$f(v_D) = V_{in} - I_S(e^{\frac{q}{kT}v_D} - 1)R - v_D$$

برنامه شما باید

۱. از عبارت `global` برای مقادیر V_{in} در تابع $f(v_D)$ استفاده کند.

۲. ولتاژ v_D در بازه از ۰.۲V تا ۰.۸V در ۶۰ قدم باشد.

۳. از روش جستجو برای پیدا کردن فواصل کوچکی که ریشه $f(v_D) = 0$ قرار دارد استفاده کنید.

۴. از تابع `fzero` متلب برای تعیین دقیق‌تر ریشه $f(v_D) = 0$ استفاده کنید.

۵. جدولی ترتیب دهید که تمام مقادیر V_{in} و ریشه‌های مربوطه $f(v_D) = 0$ را داشته باشد.

۶. اگر پروژه ۱.۴ را انجام داده‌اید، برای تأیید برنامه خود، وقتی $V_{in} = 5V$ است، باید همان جواب‌های پروژه ۱.۴ باشد.

۳.۴ پروژه

پروژه ۲.۴ را تکرار کنید، اما در این لحظه، از عبارت `global` استفاده نکرده و بجای آن تابع $f(v_D)$ را اصلاح کنید بطوری که آرگومان V_{in} را شامل شود. در این حالت لازم است تابع `fzero` را بصورت یک تابع با چند آرگومان اصلاح کنید.

۴.۴ پروژه

میخواهیم زمان استقرار^۴ (که در زیر تعریف می‌شود) را برای جریان i_L یک سیم‌پیچ در یک مدار RLC بخش ۹.۲ و پروژه ۱۷.۲ ملاحظه نمائیم. مقاومت R_{crit} که مدار میراثی بحرانی پیدا می‌کند چنین است

$$R_{crit} = \sqrt{\frac{L}{4C}}$$

برای $R > R_{crit}$ ، معادله حاکم بر i_L برابر است با

$$i_L(t) = \exp\left(-\frac{1}{2RC}t\right) \left[A \cos\left(\sqrt{\frac{1}{LC} - (\frac{1}{2RC})^2}t\right) + B \sin\left(\sqrt{\frac{1}{LC} - (\frac{1}{2RC})^2}t\right) \right]$$

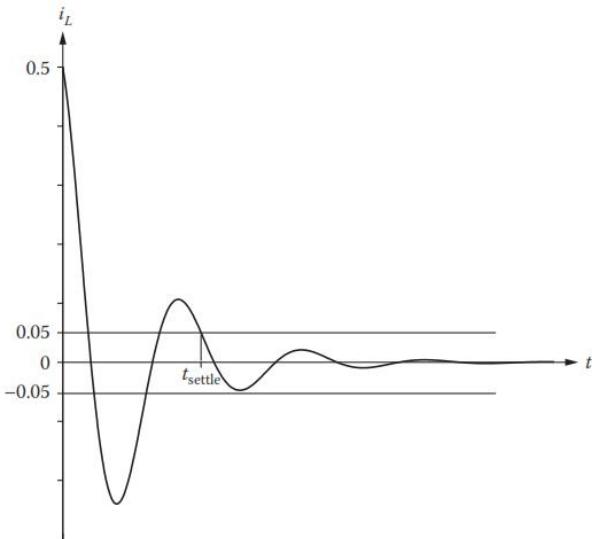
^۴settling time

ضرایب A و B از شرایط اولیه و مقادیر عناصر بدست می‌آید. جواب A و B عبارتند از

$$A = i_L(0)$$

$$B = \frac{1}{\sqrt{\frac{1}{LC} - (\frac{1}{2RC})^2 t}} \times \frac{v(0)}{L} + \frac{i_L(0)}{2RC}$$

زمان استقرار بک شکل موج مدت زمان t_{settle} است که طی می‌شود تا شکل موج (i_L) در محدوده



شکل ۵.۴: زمان استقرار t_{settle} زمانی است که طی می‌شود تا i_L با مقداری خطای معین به مقدار نهائی خود برسد. در این مسئله با محدوده خطای $\pm 10\%$ استفاده می‌شود، اما محدوده خطای $\pm 5\%$ و $\pm 2\%$ هم مرسم مرسوم است.

مقدار درصد خاصی قرار گیرد و در همان بازه باقی بماند(به شکل ۵.۴ مراجعه کنید). برای این مسئله، زمان استقرار $i_L(t)$ را در بازه ده درصد مقدار اولیه $i_L(0)$ در نظر گرفته و در همان مقدار باقی می‌ماند. برنامه‌ای بزیان متلب نوشته بطوری که زمان استقرار را برای پارامترهای زیر تعیین نماید:

$$R = 4R_{crit}, \quad L = 1.0mH, \quad C = 1.0\mu F, \quad i_L(0) = 0.25A, \quad v(0) = 6V, \quad 0 \leq t \leq 1000\mu s$$

۱. منحنی $i_L(t)$ را بر حسب زمان t رسم کنید. روی همان منحنی، خطوط $0.1i_L(0)$ و $-0.1i_L(0)$ را رسم نمائید.

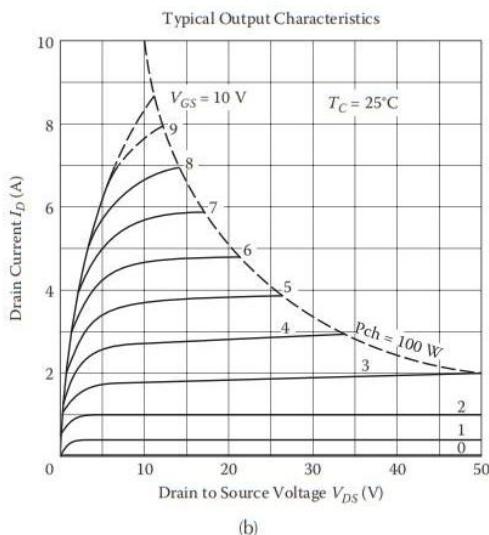
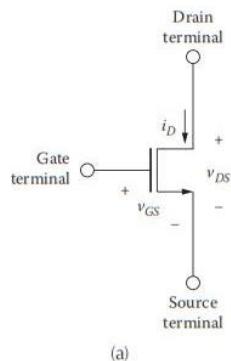
۲. با استفاده از روش جستجو، و تابع `fzero` متلب، زمان استقرار $i_L(t)$ را بدست آورید. راهنمایی: از $|i_L(t)|$ و خط $0.1i_L(0)$ استفاده کرده و ریشه بیشینه را تعیین کنید. در این صورت زمان استقرار بدست می‌آید.

۵.۴ پروژه

در شکل (۵.۴)(الف) یک کانال نیمه‌هادی فلز-اکسید ترانزیستور اثر میدان (MOSFET) نشان داده شده است. این ماسفت دارای سه پایانه: گیت (دروازه)^۵ منبع^۶ و درین (چاک)^۷ است. (در

^۵gate
^۶source
^۷drain

واقع، ماسفت‌ها نیز دارای یک پایانه دیگری بنام بالک^۱ هستند که در اینجا بحث نمی‌شود). در مدارهای ماسفت، ما معمولاً جریان درین i_D را بر جسب تابعی از ولتاژ پایانه درنظر می‌گیریم. مدل ریاضیات مرسوم برای جریان i_D درین برای ماسفت کانال n بقرار زیر است [۶]:



شکل ۴.۶: (الف) کانال نوع N ترانزیستور MOS و تعریف پارامترهای پایانه. (ب) مشخصات پایانه ماسفت قدرت $.2SK1056$

$$i_D = \begin{cases} 0 & v_{GS} \leq V_{th} \\ K [2(v_{GS} - V_{th})V_{DS} - V_{DS}^2(1 - \lambda V_{DS})] & v_{GS} > V_{th} \quad \text{و} \quad v_{DS} \leq v_{GS} - V_{th} \\ K(v_{GS} - V_{th})^2(1 - \lambda V_{DS}) & v_{GS} > V_{th} \quad \text{و} \quad v_{DS} > v_{GS} - V_{th} \end{cases} \quad \text{برای } \begin{array}{l} v_{GS} \leq V_{th} \\ v_{GS} > V_{th} \quad \text{و} \quad v_{DS} \leq v_{GS} - V_{th} \\ v_{GS} > V_{th} \quad \text{و} \quad v_{DS} > v_{GS} - V_{th} \end{array}$$

(۹.۴)

که در آن i_D در شکل (۶.۴)(الف) تعریف شده است؛ v_{GS} و V_{DS} بترتیب ولتاژهای منبع-چاهک و دروازه-چاهک ترانزیستور هستند؛ V_{th} "ولتاژ آستانه"^۹ ترانزیستور و K و λ ثابت‌هایی هستند که بستگی به سطح ناچالصی، تحرک الکترونها^{۱۰} و شکل هندسی قطعه دارد. شکل (۶.۴)(ب)

^۱bulk

^۹threshold voltage

^{۱۰}electron mobility

نمودار i_D را بر حسب تابعی از V_{DS} (بازاء مقدار ثابت v_{GS}) برای ماسفت قدرت 2SK1056 نشان می‌دهد.

۱. تابعی در مطلب بنام $id_{powerfet}$ بنویسید که دو آرگومان v_{ds} و v_{gs} را گرفته و جریان i_d را که طبق رابطه [۹.۴](#) تعریف شده است محاسبه نماید. در اینجا $V_{th} = 0.2V$, $K = 0.2$, $\lambda = 0.01$

در نظر گیرید.

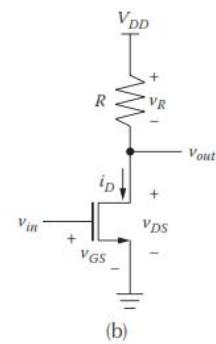
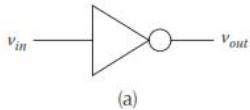
۲. جریان i_D را بر حسب v_{DS} را محدوده صفر تا پنجاه ولت بازاء $v_{GS} = 1, 2, 3, \dots, 8$ رسم کنید. شکل شما باید مشابه شکل [\(۶.۴\)\(ب\)](#) شود.

۳. چون 2SK1056 برای حداکثر توان صد وات طرح شده است، شکل [\(۶.۴\)\(ب\)](#) وقتی جریان i_D مقداری است که $P = i_D \times v_{DS}$ بیش از صد وات است، نشان داده نشده است. تابع خود را بقسمی اصلاح کنید که $iD_{powerfet}$ آزمایش کند که آیا این در محدوده توان قرار دارد و در غیر این صورت جواب NaN دهد. اکنون نمودار مشابه شکل [\(۶.۴\)\(ب\)](#) شود.

۴. با استفاده از تابع $fzero$ مقدار v_{GS} را وقتی $i_D = 2 A$ و $v_{DS} = 5 V$ است بدست آورید. راهنمائی: برای این مسئله لازم است تا $iD_{powerfet}$ را بر حسب تابعی از v_{GS} با ثابت گرفتن i_D و v_{DS} حل کنید.

۶.۴ پروژه

ساده‌ترین گیت منطقی دیژیتالی معکوس کننده^{۱۱}، شکل [\(۷.۴\)](#)، است که "صحیح" و "غلط" جبر بول^{۱۲} را جایشان را عوض می‌کند. در الکترونیک، معمولاً ما "صحیح" منطقی را با ولتاژ زیاد (مثلاً پنج ولت) و "غلط" منطقی را با ولتاژ کم (مثلاً صفر ولت) نشان میدهیم. بنابراین، از



شکل ۷.۴: (الف) معکوس کننده منطقی (ب) مدار معکوس کننده (RTL) با استفاده از ترانزیستور ماسفت کanal n

نقشه نظر الکترونیک، یک معکوس کننده مداری است که ورودی $+5V$ را گرفته و در خروجی صفر ولت و بر عکس صفر ولت ورودی را گرفته و در خروجی $+5V$ میدهد. مدار ساده‌ای که این ملزمات را داشته باشد یک مدار ماسفت مقاومتی شکل [\(۷.۴\)\(ب\)](#) و آن بصورت منطق

^{۱۱}inverter

^{۱۲}Boolean

مقاومت-ترانزیستور (*RTL*)^{۱۳} است. در این حالت، یک ماسفت کanal n که با مقاومت دو کیلو اهم بار شده و ولتاژ منبع تغذیه V_{DD} بوده (معمولاً پنج ولت)، و v_{in} ولتاژ ورودی (آن برابر v_{GS} است)، و ولتاژ خروجی v_{out} (آن برابر v_{DS} است) این کار را انجام میدهد. با استفاده از قانون ولتاژ کیرشوف، میتوانیم ولتاژ حلقه خروجی را بنویسیم

$$V_{DD} - v_R - v_{DS} = 0 \quad (10.4)$$

$$v_R = i_D R \quad (11.4)$$

$$V_{DD} - i_D R - v_{DS} = 0 \quad (12.4)$$

که در آن i_D بر طبق رابطه (۹.۴) بدست می‌اید.

برنامه‌ای بزبان متلب نوشته بطوری که ازتابع *fzero* متلب استفاده و ریشه‌های معادله (۱۲.۴) را محاسبه کند. برای ماسفت کanal n از پارامترهای زیر استفاده شود:

$$V_{DD} = 5 \text{ V}, \quad V_{th} = 0.7 \text{ V}, \quad K = 0.0005, \quad \lambda = 0.05$$

مقدار $0 \leq v_{DS} \leq 5 \text{ V}$ در قدمهای $0.1 \text{ V} \leq v_{GS} \leq 5 \text{ V}$ در نظر گیرید. در واقعیت امر، بازه ولتاژها را سیگنال "پائین" و "بالا" بحساب می‌آورند. برای معکوس کننده (*RTL*) که در اینجا بحث شد، فرض کنید بازه $1 \text{ V} - 0$ تشکیل سیگنال "پائین" و $4 - 5 \text{ V}$ سیگنال "بالا" است

۱. جدولی درست کنید که v_{out} را بر حسب v_{in} (مربوط به v_{DS} در مقابل v_{GS}) بما دهد.

۲. نمودار v_{out} را در مقابل v_{in} رسم کنید.

۳. آیا مداری که در اینجا بحث شد می‌تواند مانند یک معکوس کننده عمل کند، یعنی وقتی ورودی بالا است خروجی پائین تولید کند و برعکس؟

۴. مقدار نویز حاشیه‌ای^{۱۴} بصورت مقدار نویزی که می‌تواند در وروری داشته باشد بنهوی که خروجی ثابت باقی بماند. برای مثال، اگر v_{in} به حد معقول پائین بودن، مثلاً به 1.1 V افزایش یابد، آیا خروجی در همان مقدار بالا معتبر باقی خواهد ماند؟ بیشترین مقدار ممکن را برای ورودی سطح پائین v_{in} بدست آورید تا خروجی v_{out} در همان سطح بالا باقی بماند. ضمناً، کمترین مقدار سطح بالا ورودی v_{in} چقدر باشد تا همان خروجی سطح پائین v_{out} را داشته باشیم. این‌ها را روی صفحه نمایشگر نشان دهید.

۷.۴ پروژه

در مدار منطقی *RLT* پروژه ۶.۴ در سال ۱۹۵۰ وقتی ترانزیستور گران بود و میباشد تک تک بسته بندی شوند، بکار برد می‌شد و مقاومت هم ارزان بود. وقتی مدارات مجتمع در سال ۱۹۶۰ اختراع شد، قضیه معکوس شد، ترانزیستور ارزان و مقاومت نسبتاً گران شد (زیرا تعییه آن روی تراشه^{۱۵} مشکل است).

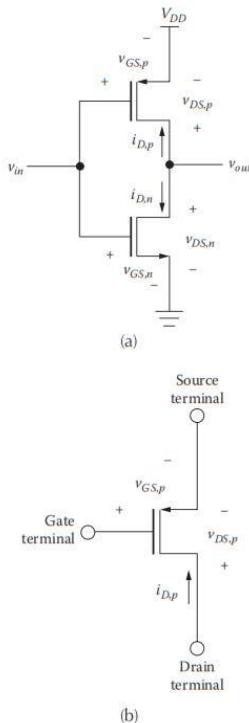
شکل (۸.۴)(الف) متمم^{۱۶} معکوس کننده یک ماسفت (سی‌ماس) MOSFET(CMOS) را نشان میدهد. در این حالت، مقاومت با کanal p ماسفت جایگزین شده است. از نظر ریاضی می‌توانیم ماسفت کanal p را شبیه ماسفت کanal n پروژه ۵.۴ مدل سازی کنیم؛ یعنی، جریان دروازه کanal p چنین است.

^{۱۳}resistor-transistor logic

^{۱۴}noise margin

^{۱۵}chip

^{۱۶}complementary



شکل ۸.۴: (الف) مدار معکوس کننده سیماس (ب) ترانزیستور ماس کانال p و پارامترهای تعریف شده پایانه‌های آن.

$$i_{D,P} = \begin{cases} 0 & v_{GS,P} \geq V_{th,P} \\ -K_P [2(v_{GS,P} - V_{th,P})v_{DS,P} - v_{DS,P}^2(1 - \lambda_P v_{DS,P})] & v_{GS,P} < V_{th,P} \quad \text{و} \quad v_{DS,P} \geq (v_{GS,P} - V_{th,P}) \\ -K_P(v_{GS,P} - V_{th,P})^2(1 - \lambda v_{DS,P}) & v_{GS,P} < V_{th,P} \quad \text{و} \quad v_{DS,P} < (v_{GS,P} - V_{th}) \end{cases} \quad (13.4)$$

که در آن $i_{D,P}$ ، $v_{GS,P}$ ، $v_{DS,P}$ و $V_{th,P}$ در شکل (۸.۴) تعریف شده‌اند. توجه کنید که مدل کانال p شبیه مدل کانال n رابطه (۹.۴) با این تفاوت که علامت‌ها عوض می‌شود. بعلاوه، برای فناوری مدار مجتمع، مقادیر K ، V_{th} و λ برای نوع p متفاوت از کانال ماسفت نوع n است (ولذا بترتیب با اندیس p و n نشان داده می‌شوند). قانون ولتاژ کیرشف را برای ارتباط ولتاژ سمت خروجی مدار بکار می‌بریم:

$$v_{DS,n} = v_{out} \quad (14.4)$$

$$v_{GS,P} = v_{out} - V_{DD} \quad (15.4)$$

به همین نحو، در سمت ورودی:

$$v_{DS,n} = v_{in} \quad (16.4)$$

$$v_{GS,P} = v_{in} - V_{DD} \quad (17.4)$$

ضمناً قانون جریان کیرشوف را در گره خروجی اعمال می‌کنیم:
 $i_{D,n} + i_{D,p} = 0 \quad (18.4)$

که در آن جریان $i_{D,n}$ منبع کanal n بصورت زیر تعریف می‌شود

$$i_{D,n} = \begin{cases} 0 & v_{GS,n} \leq V_{th,n} \\ K_n [2(v_{GS,n} - V_{th,n})v_{DS,n} - v_{DS,n}^2(1 - \lambda_n v_{DS,n})] & v_{GS,n} > V_{th,n} \text{ و } v_{DS,n} \leq (v_{GS,n} - V_{th,n}) \\ K_n(v_{GS,n} - V_{th,n})^2(1 - \lambda v_{DS,n}) & v_{GS,n} > V_{th,n} \text{ و } v_{DS,n} > (v_{GS,n} - V_{th,n}) \end{cases} \quad (19.4)$$

- برنامه‌ای بزیان متلب با استفاده از تابع `fzero` بنویسید که ریشه‌های معادله (۱۸.۴) را با اعمال معادلات (۱۹.۴) و (۱۳.۴) بدست آورد. برای برنامه از پارامترهای زیر استفاده شود

$$\begin{aligned} V_{DD} &= 5 \text{ V}, \quad V_{th,n} = 0.7 \text{ V}, \quad V_{th,p} = -0.8 \text{ V}, \\ K_n &= 0.0005, \quad K_p = 0.00015, \quad \lambda_n = 0.05, \quad \lambda_p = 0.1 \end{aligned}$$

محدوده ولتاژ خروجی v_{out} را از صفر ولت تا پنج ولت در قدمهای ۰.۱ V و بازه ولتاژ ورودی v_{in} را از صفر ولت تا پنج ولت در قدمهای ۰.۱ V در نظر گیرید.

- جدولی برای v_{out} در مقابل v_{in} تولید کنید.
- نمودار v_{out} در مقابل v_{in} رارسم کنید.
- آیا این مدار شبیه یک مدار معکوس کننده عمل می‌کند؟ یعنی ورودی بالا است خروجی پائین تولید کند و برعکس؟

فصل ۵

انتگرال گیری عددی

۱.۵ مقدمه

در این فصل، قانون سیمپسون^۱ را برای حل تقریبی انتگرال‌ها معین بدهست می‌آوریم. سپس استفاده آنرا در توابع متلب quad و dblquad برای محاسبه انتگرال‌های معین نشان خواهیم داد. در نهایت، این توابع را در حل مسائل معمولی نظریه میدان‌های الکترومغناطیسی بکار خواهیم برد.

۲.۵ انتگرال گیری عددی و قانون سیمپسون

میخواهیم انتگرال I زیر را محاسبه کنیم

$$I = \int_A^B f(x) dx \quad (1.5)$$

مراحل برای محاسبه عددی I بصورت زیر است:

■ محور x ها را از A تا B به $x = B$ تا $x = A$ قسمت تقسیم می‌کنیم، بطوریکه N یک عدد زوج باشد.

■ قانون سیمپسون شامل اتصال گروه سه نقطه‌ای روی منحنی $f(x)$ با چندجمله‌ای درجه دوم (سهمی) بوده و جمع سطح زیر منحنی سهمی‌ها برای بدست آوردن سطح تقریبی زیر منحنی اصلی است (به شکل ۱.۵ مراجعه کنید).

ابتداتابع $f(x)$ را بر حسب سری تیلور^۲ حول نقطه x_i با استفاده از سه جمله، یعنی

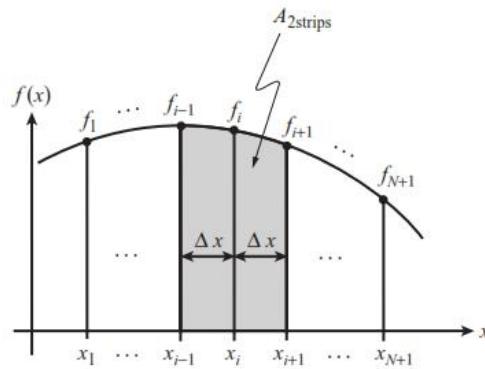
$$f(x) = a(x - x_i)^2 + b(x - x_i) + c \quad (2.5)$$

که در آن a ، b و c را باید بدست آوریم، بسط میدهیم.
سپس، سطح زیر منحنی دو نوار نزدیک بهم $A_{2strips}$ را با انتگرال گیری بین x_{i-1} و x_{i+1} محاسبه می‌نماییم:

$$A_{2strips} = \int_{x_{i-1}}^{x_{i+1}} f(x) dx = \int_{x_i - \Delta x}^{x_i + \Delta x} [a(x - x_i)^2 + b(x - x_i) + c] dx \quad (3.5)$$

¹impson's rule

²Taylor series

شکل ۱.۵: سه نقطه دلخواه روی منحنی $f(x)$

فرض کنید $\xi = x - x_i$ باشد، در این صورت $d\xi = dx$ وقتی $d\xi = \Delta x$ $x = x_i + \Delta x$ و وقتی $d\xi = -\Delta x$ $x = x_i - \Delta x$ است. با قراردادن اینها در رابطه (۳.۵) خواهیم داشت

$$\begin{aligned} A_{2strip} &= \int_{x_{i-1}}^{x_{i+1}} f(x) dx \\ &= \int_{-\Delta x}^{\Delta x} (a\xi^2 + b\xi + c) d\xi \\ &= \left. \frac{a\xi^3}{3} + \frac{b\xi^2}{2} + c\xi \right|_{-\Delta x}^{\Delta x} \\ &= \frac{a}{3}(\Delta x)^3 - \frac{a}{3}(-\Delta x) + \frac{b}{2} \frac{a}{3}(\Delta x)^2 - \frac{b}{2} \frac{a}{3}(\Delta x)^2 + c(\Delta x) - c(-\Delta x) \end{aligned}$$

بنابراین

$$A_{2strip} = \frac{2a}{3}(\Delta x)^3 + 2c(\Delta x)$$

اکنون، تابع $f_i = f(x_i)$ را تعریف می‌کنیم. با حل معادله (۲.۵) برای f_i, f_{i-1} و f_{i+1} خواهیم داشت

$$f(x_i) = f_i = c$$

$$f(x_{i+1}) = f_{i+1} = a(\Delta x)^2 + b(\Delta x) + c$$

$$f(x_{i-1}) = f_{i-1} = a(-\Delta x)^2 + b(-\Delta x) + c$$

با اضافه کردن دو معادله اخیر داریم $a \cdot f_{i+1} + f_{i-1} = 2a\Delta x^2 + 2c$ و حل آن برای a می‌دهد

$$a = \frac{1}{2\Delta x^2} [f_{i+1} + f_{i-1} - 2f_i]$$

در این صورت،

$$A_{2strip} = \frac{2}{3} \times \frac{1}{2\Delta x^2} [f_{i+1} + f_{i-1} - 2f_i] (\Delta x)^3 + 2f_i \Delta x = \frac{\Delta x}{3} [f_{i+1} + f_{i-1} - 2f_i + 6f_i]$$

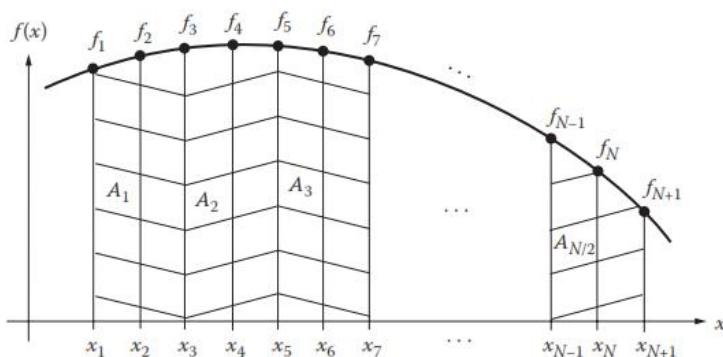
یا

$$A_{2strip} = [f_{i+1} + 4f_i + f_{i-1}] \quad (۴.۵)$$

برای تعیین مقدار تقریبی انتگرال I ، لازم است تا تمام سطح دو نوار زیر منحنی از $x = A$ تا $x = B$ را جمع کنیم (به شکل ۱.۵ مراجعه کنید). یعنی

$$\begin{aligned} A_1 &= \frac{\Delta x}{3} [f_1 + 4f_2 + f_3] \\ A_2 &= \frac{\Delta x}{3} [f_3 + 4f_4 + f_5] \\ A_3 &= \frac{\Delta x}{3} [f_5 + 4f_6 + f_7] \\ &\vdots \\ A_{N/2} &= \frac{\Delta x}{3} [f_{N-1} + 4f_N + f_{N+1}] \end{aligned}$$

بنابراین،



شکل ۲.۵: انتگرال‌گیری سطح توسط قانون سیمپسون

$$I = \int_{x_1=A}^{x_{N+1}=B} f(x) dx = \frac{\Delta x}{3} [f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \cdots + 4f_N + f_{N+1}] \quad (۵.۵)$$

این قانون سیمپسون برای انتگرال‌گیری است.

مثال ۱.۵

مقدار انتگرال زیر را محاسبه کنید:

$$I = \int_0^{10} (x^3 + 3.2x^2 - 3.4x + 20.2) dx$$

```
% Example_5_1.m
% This program calculates an integral by Simpson's Rule
% The integrand is: x^3+3.2*x^2-3.4*x+20.2
% The limits of integration are from 0-10.
clear; clc;
A=0; B=10;
N=100; dx=(B-A)/N;
% Compute values of x and f at each point:
x = A:dx:B;
f = x.^3 + 3.2*x.^2 - 3.4*x + 20.2;
% Use two separate loops to sum up the even and odd terms
% of Simpson's Rule. Also, exclude endpoints in the loop.
```

```

sum_even=0.0;
for i=2:2:N
sum_even=sum_even+f(i);
end
sum_odd=0.0;
for i=3:2:N-1
sum_odd=sum_odd+f(i);
end
% Calculate integral as per Equation 5.5
I = dx/3 * (f(1) + 4*sum_even + 2*sum_odd + f(N+1));
% Display results
fprintf('Integrand: x^3 + 3.2*x^2 -3.4*x + 20.2 \n');
fprintf('Integration limits: %.1f to %.1f \n',A,B);
fprintf('I = %10.4f \n',I);
%-----

```

و جواب برنامه:

```

Integrand: x^3 + 3.2*x^2 -3.4*x + 20.2
Integration limits: 0.0 to 10.0
I = 3598.6667

```

۳.۵ انتگرال‌های نامعین

یک انتگرال نامعین وقتی نامعین است که وقتی زیرانتگرال آن بسمت در هر یک از کرانه‌هایش میل کند به نهایت شود. در خیلی از حالات، انتگرال جواب معینی دارد.

مثال ۲.۵:

$$I = \int_0^1 \frac{\log(1+x)}{x} dx \quad (6.5)$$

این انتگرال نامعین است زیرا هر دو صورت و مخرج کسر در حد پائین ($x = 0$) صفر هستند. مقدار دقیقی I را می‌توان با استفاده از نظریه باقیمانده‌ها^۶ در متغیرهای مختلط بدست آورد، و در این حالت مقدار آن برابر $I = \frac{\pi^2}{12} = 0.822467$ است. فرض کنید

$$I = \int_0^1 \frac{\log(1+x)}{x} dx = I_1 + I_2$$

که در آن

$$I_1 = \int_{\epsilon}^1 \frac{\log(1+x)}{x} dx$$

$$I_2 = \int_0^{\epsilon} \frac{\log(1+x)}{x} dx$$

^۶residue theory

در اینجا $\epsilon \ll 1$ میباشد. برای محاسبه I_2 تابع $\log(1+x)$ را بر حسب سری تیلور حول نقطه $x = 0$ بسط میدهیم، داریم

$$\log(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \frac{1}{5}x^5 - \dots + \dots$$

در این صورت

$$\frac{\log(1+x)}{x} = 1 - \frac{1}{2}x + \frac{1}{3}x^2 - \frac{1}{4}x^3 + \frac{1}{5}x^4 - \dots + \dots$$

$$I_2 = \int_0^\epsilon \left(1 - \frac{1}{2}x + \frac{1}{3}x^2 - \frac{1}{4}x^3 + \frac{1}{5}x^4 - \dots + \dots\right) dx$$

$$I_2 = \epsilon - \frac{1}{4}\epsilon^2 + \frac{1}{9}\epsilon^3 - \frac{1}{16}\epsilon^4 + \frac{1}{25}\epsilon^5 - \frac{1}{36}\epsilon^6 + \frac{1}{49}\epsilon^7 - \dots + \dots \quad (7.5)$$

مقدار I_1 را با استفاده از قانون سیمپسون و I_2 را با استفاده از رابطه (۷.۵) محاسبه می‌نمائیم. برنامه زیر این روش را نشان میدهد:

```
% Example_5_2.m
% This program evaluates the improper integral:
% log(1+x)/x The limits of integration are from 0 to 1.
% The integrand is undefined (0/0) at x=0. Thus, the
% integral is broken up into 2 parts: I1 and I2.
% I1 is evaluated from epsilon to 1.
% I2 is expanded in a Taylor Series and evaluated from 0
% to epsilon.
clear; clc;
epsilon = 0.0001;
A = 0; B = 1;
N = 100; dx = (B-epsilon)/N;
% Evaluate I1. First, evaluate log(1+x)/x for each value
% of x over the interval [epsilon,1].
x = epsilon:dx:B;
f = log(1+x) ./ x;
% Next, calculate the even and odd terms for Simpson's
% Rule.
sumeven=0.0;
for i=2:2:N
    sumeven=sumeven+f(i);
end
sumodd=0.0;
for i=3:2:N-1
    sumodd=sumodd+f(i);
end
% As per Equation (5.5), I1 is the weighted sum of the
% even and odd terms, plus the end terms.
I1 = dx/3 * (f(1) + 4*sumeven + 2*sumodd + f(N+1));
fprintf('Integrand = log(1+x)/x \n');
fprintf('I1 limits of integration are '');
```

```

fprintf('from %.4f to %.4f\n',epsilon,B);
fprintf('I2 limits of integration are ');
fprintf('from %.4f to %.4f\n',A,epsilon);
fprintf('I1 =%15.6f \n',I1);
% Calculate I2 via first 4 terms of Taylor series
% expansion.
I2 = epsilon -1/4*epsilon^2 + 1/9*epsilon^3 ...
-1/16*epsilon^4;
fprintf('I2 =%15.6f \n',I2);
I=I1+I2;
fprintf('I=I1+I2 =%10.6f \n',I);
I_exact=0.822467;
fprintf('I_exact =%10.6f \n',I_exact);
-----

```

و جواب برنامه:

```

Integrand = log(1+x)/x
I1 limits of integration are from 0.0001 to 1.0000
I2 limits of integration are from 0.0000 to 0.0001
I1 = 0.822367
I2 = 0.000100
I = I1+I2 = 0.822467
I_exact = 0.822467

```

بطوری که مشاهده می‌کنید این روش کاملاً جواب خوبی می‌دهد.

۴.۵ تابع quad متلب

تابع متلب برای محاسبه یک انتگرال `quad` است. توضیح تابع را می‌توان با نوشتتن `help quad` در پنجره فرمان بدست آورد. مطالب آن در زیر آورده می‌شود:

```

Q = QUAD(FUN,A,B) tries to approximate the integral of scalarvalued function
FUN from A to B to within an error of 1.e-6
using recursive adaptive Simpson quadrature. FUN is a function
handle. The function Y = FUN(X) should accept a vector argument
X and return a vector result Y,the integrand evaluated at each
element of X.
Q = QUAD(FUN,A,B,TOL) uses an absolute error tolerance of TOL
instead of the default,which is 1.e-6. Larger values of TOL
result in fewer function evaluations and faster computation,
but less accurate results. The QUAD function in MATLAB 5.3 used
a less reliable algorithm and a default tolerance of 1.e-3.

```

بنابراین، تابع آرگومان `quad` که بصورت ام‌فایل *m*. تعریف شده است را به عنوان زیرانتگرال همراه کرانه‌های انتگرال قبول می‌کند. روش دیگر این که اگر زیر انتگرال خیلی متن بزرگی نداشته باشد را بصورت متن بی‌نام یا تابع همراه نوشت (به مثال ۴.۵ و ۵.۵ مراجعه کنید).

اگر زیرانتگرال مقادیر بسیار بزرگ یا شامل شود، می‌توانید تولرانس در آرگومان سوم quad یعنی تولرانس پیش فرض را تغییر دهید (این در توضیح متن دوم آورده شده است). تابع quad می‌تواند انتگرال‌های نامعین را نیز محاسبه نماید. (به تمرین‌های ۵.۲e و ۵.۲f مراجعه کنید).

مثال ۳.۵:

اکنون می‌خواهیم مثال ۱.۵ را تکرار کنیم، اما این دفعه برای انتگرال‌گیری از تابع quad مطلب استفاده خواهیم کرد. انتگرال I مثال ۱.۵ چنین است:

$$I = \int_0^{10} (x^3 + 3.2x^2 - 3.4x + 20.2) dx$$

و برنامه آن بقرار زیر است:

```
% Example_5_3.m
% This program evaluates the integral of the function
% 'f1' between A and B by MATLAB's quad function. Since
% the function 'f1' is just a single line, we can use the
% anonymous form of the function.
clear; clc;
f1=@(x) (x.^3+3.2*x.^2-3.4*x+20.2);
A=0.0; B=10.0;
I = quad(f1,A,B);
fprintf('Integration of f1 over [% .0f, % .0f] ',A,B);
fprintf('by MATLAB''s quad function:\n');
fprintf('f1 = x^3 + 3.2*x^2 -3.4*x + 20.2 \n');
fprintf('integral = %10.4f \n',I);
%-----
```

و جواب برنامه:

```
Integration of f1 over [0,10] by MATLAB's quad function:
f1 = x^3 + 3.2*x^2 -3.4*x + 20.2
integral = 3598.6667
```

بطوری که مشاهده می‌کنید این با جوابی که در مثال ۱.۵ بدست آمد یکسان است.

مثال ۴.۵:

مطلوبست محاسبه

$$\int_0^1 \frac{t}{t^3 + t + 1} dt$$

```
% Example_5_4.m
% This program evaluates the integral of function "f2"
% between A and B by MATLAB's quad function. The
% integrand can be expressed in a single line and thus we
% can use the anonymous form of the function.
clear; clc;
A=0.0; B=1.0;
```

```
f2 = @(t) t ./ (t.^3 + t + 1.0);
I2 = quad(f2,A,B);
fprintf('Integration of f2 over [% .0f,% .0f] ',A,B);
fprintf('by MATLAB''s quad function:\n');
fprintf('f2 = t/(t^3 + t + 1) \n');
fprintf('integral=%f \n',I2);
%-----
```

و جواب برنامه:

```
Integration of f2 over [0,1] by MATLAB's quad function:
f2 = t/(t^3 + t + 1)
integral = 0.260068
```

بیایید مثال 2.5 را تکرار کنیم، بطوری که انتگرال نامعین معادله (۶.۵) مورد نظر باشد، اما این دفعه آنرا با استفاده از تابع quad در کرانه‌های یک تا صفر محاسبه نمائیم. بیاد آورید که تابع $\log(1+x)/x$ در صفر نامعین است. بعلاوه، میخواهیم از تابع بنام استفاده کنیم.

```
% Example_5_5.m
% This program evaluates the improper integral log(1+x)/x
% with limits from 0 to 1 using quad.
clear; clc;
A=0; B=1;
fprintf('This program uses the quad function to\n');
fprintf('evaluate the integral of log(1+x)/x from ');
fprintf('%.0f to %.0f.\n',A,B);
I = quad(inline('log(1+x)./x'),A,B);
fprintf('I = %10.6f \n',I);
%-----
```

و جواب برنامه:

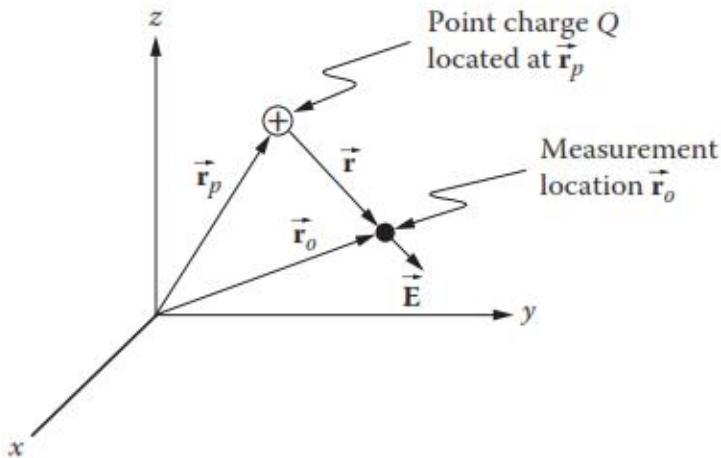
```
This program uses the quad function to evaluate
the integral of log(1+x)/x from 0 to 1.
I = 0.822467
```

این جواب نیز با جواب مثال 2.5 یکسان است.

۵.۵ میدان الکتریکی

مسائل میدان‌های الکتریکی اغلب انتگرال را نیز در بر داشته، و قبل از آنکه اقدامی کنیم، برخی از مفاهیم اساسی میدان‌های الکترومغناطیسی را مرور می‌کنیم.
فرض کنید بار نقطه‌ای Q کولمب در مختصات (x_p, y_p, z_p) در فضای آزاد قرار دارد. ما محل Q را با بردار \vec{r}_p (به شکل ۳.۵ مراجعه کنید) مشخص می‌کنیم. میدان الکتریکی \vec{E} در نقطه مشاهده \vec{r}_o مربوط به مختصات (x_o, y_o, z_o) ناشی از بار نقطه واقع در \vec{r}_p چنین تعریف می‌شود:

$$\vec{E} = \frac{Q}{4\pi\epsilon_0|\vec{r}|^2}\hat{e}_r \quad (8.5)$$

شکل ۳.۵: میدان الکتریکی \vec{E} در نقطه مشاهده \vec{r}_o ناشی از بار نقطه Q واقع در \vec{r}_p

که در آن ϵ_0 گذردهی الکتریکی فضای آزاد ($8.85 \times 10^{-12} \text{ farad/m}$), $\hat{\mathbf{e}}_r$ برداری از بار نقطه‌ای به نقطه اندازه‌گیری، $\hat{\mathbf{e}}_r$ بردار واحد در جهت \vec{r} و \vec{E} میدان بر حسب ولت بر متر است. از شکل (۳.۵) می‌توان جمع برداری $\vec{r} = \vec{r}_o - \vec{r}_p$ را نوشت و در نتیجه $\vec{r} = \vec{r}_o + \vec{r}_p$ را بدست آورد. بنابراین معادله (۸.۵) خواهد شد

$$\vec{E} = \frac{Q}{4\pi\epsilon_0 |\vec{r}_o - \vec{r}_p|^2} \frac{\vec{r}_o - \vec{r}_p}{|\vec{r}_o - \vec{r}_p|} \quad (9.5)$$

که در آن بردار واحد را بصورت $\hat{\mathbf{e}}_r = \frac{\vec{r}}{|\vec{r}|}$ تعریف کردیم.
اکنون \vec{r}_o و \vec{r}_p را بر حسب مولفه‌های x , y , z در مختصات دکارتی بصورت $\vec{r}_o = x_o \hat{\mathbf{e}}_x + y_o \hat{\mathbf{e}}_y + z_o \hat{\mathbf{e}}_z$ و $\vec{r}_p = x_p \hat{\mathbf{e}}_x + y_p \hat{\mathbf{e}}_y + z_p \hat{\mathbf{e}}_z$ می‌نویسیم. در اینجا $\hat{\mathbf{e}}_x$, $\hat{\mathbf{e}}_y$ و $\hat{\mathbf{e}}_z$ بردارهای واحد نرتیب در راستای x , y و z هستند. در این صورت، $|\vec{r}_o - \vec{r}_p| = \sqrt{(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2}$ رابطه (۸.۵) خواهد شد

$$\vec{E} = \frac{Q}{4\pi\epsilon_0} \frac{(x_o - x_p)\hat{\mathbf{e}}_x + (y_o - y_p)\hat{\mathbf{e}}_y + (z_o - z_p)\hat{\mathbf{e}}_z}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (10.5)$$

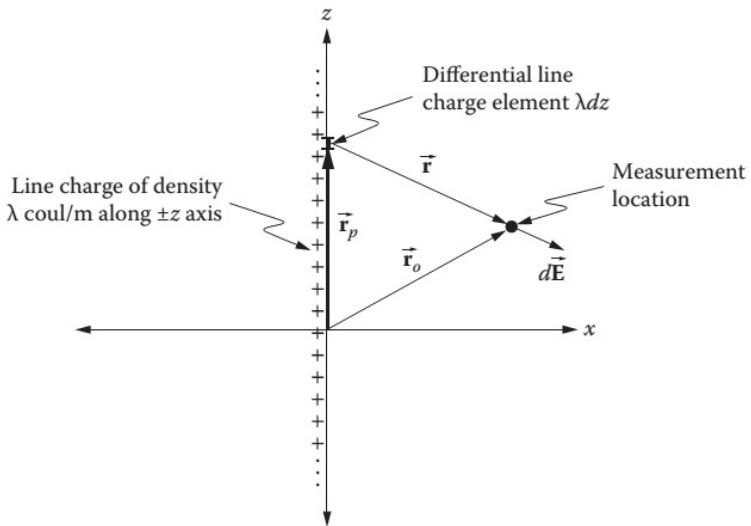
بنابراین، رابطه (۱۰.۵) میدان الکتریکی را در نقطه مشاهده (x_o, y_o, z_o) در اثر بار نقطه‌ای Q واقع در (x_p, y_p, z_p) بما میدهد.

اکنون می‌خواهیم میدان الکتریکی ناشی از یک خط بردار در روی محور z ها از $-\infty$ تا $+\infty$ (به شکل ۴.۵ مراجعه کنید) محاسبه کنیم. فرض کنید بار در روی خط با چگالی بار خطی λ کولمب بر متر توزیع شده است. می‌توان معادله (۱۰.۵) را بصورت فرم دیفرانسیلی برای $d\vec{E}$ بر حسب کمیت دیفرانسیلی $dQ = \lambda dz_p$ بازنویسی کرد. در اینجا dz_p نمایشگر قطعه بسیار کوچک بار خطی در روی محور z ها است:

$$d\vec{E} = \frac{\lambda dz_p}{4\pi\epsilon_0} \frac{(x_o - x_p)\hat{\mathbf{e}}_x + (y_o - y_p)\hat{\mathbf{e}}_y + (z_o - z_p)\hat{\mathbf{e}}_z}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (11.5)$$

با جدا سازی مولفه‌ها در جهت‌های مختلف رابطه (۱۱.۵) و انتگرال‌گیری از $-\infty$ تا $+\infty$ میدهد

$$E_x = \int_{-\infty}^{\infty} \frac{\lambda dz_p}{4\pi\epsilon_0} \frac{x_o - x_p}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (12.5)$$



شکل ۴.۵

$$E_y = \int_{-\infty}^{\infty} \frac{\lambda dz_p}{4\pi\epsilon_0} \frac{y_o - y_p}{[(y_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (13.5)$$

$$E_z = \int_{-\infty}^{\infty} \frac{\lambda dz_p}{4\pi\epsilon_0} \frac{z_o - z_p}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (14.5)$$

مثال ۶.۵

خط بارداری در روی محور z ها با چگالی $\lambda = 2 \times 10^{-9} C/m$ داده شده است. میدان اکترویکی واقع در $(x_o, y_o, z_o) = (0.005, 0, 0)$ را محاسبه کنید.

```
% Example_5_6.m
% This program uses the quad function to evaluate the
% integrals of equations (5.12), (5.13) and (5.14) to
% find Ex,Ey and Ez. The following parameters are used
% in the integration: lambda=2e-9 coul/m,xo=5e-3,yo=0
% and zo=0.
% Assume that 10 is a good approximation for infinity.
clc; clear;
global xo yo zo lambda epsilon0;
lambda=2e-9; epsilon0=8.85e-12;
xo=5e-3; yo=0; zo=0;
Ex = quad(@linecharge_Ex,-10,10);
Ey = quad(@linecharge_Ey,-10,10);
Ez = quad(@linecharge_Ez,-10,10);
fprintf('E-field due to a line charge ');
fprintf('at [%5.3f,%5.3f,%5.3f]:\n',xo,yo,zo);
```

```

fprintf(' [Ex,Ey,Ez] = [%10.4e,%5.2f,%5.2f] V/m\n',...
Ex,Ey,Ez);
%-----
% linecharge_Ex.m: Function to calculate Ex at (xo,yo,zo)
% for a line charge on the z axis
function Ex = linecharge_Ex(zp)
global xo yo zo lambda epsilon0;
Ex = lambda/(4*pi*epsilon0) * xo ...
./ (xo.^2 + yo.^2 + (zo-zp).^2).^1.5;
%-----
% linecharge_Ey.m: Function to calculate Ey at (xo,yo,zo)
% for a line charge on the z axis
function Ey = linecharge_Ey(zp)
global xo yo zo lambda epsilon0;
Ey = lambda/(4*pi*epsilon0) * yo ...
./ (xo.^2 + yo.^2 + (zo-zp).^2).^1.5;
%-----
% linecharge_Ez.m: Function to calculate Ez at (xo,yo,zo)
% for a line charge on the z axis
function Ez = linecharge_Ez(zp)
global xo yo zo lambda epsilon0;
Ez = lambda/(4*pi*epsilon0) * (zo-zp) ...
./ (xo.^2 + yo.^2 + (zo-zp).^2).^1.5;
%-----

```

و جواب برنامه:

```

E-field due to a line charge at [0.005,0.000,0.000]:
[Ex,Ey,Ez] = [7.1934e+003,0.00,0.00] V/m

```

۶.۵ رسم نمودار توسط quiver

رسم نمودار توسط دستور `quiver` یک ترسیم دوبعدی تابع برداری است که از بردارهای کوچک برای نشاندادن مقدار بردار محلی و جهت آن استفاده می‌شود. برای مثال، نمودار `quiver` می‌تواند رسم میدان الکتریکی و مغناطیسی (یعنی خروجی یک سکل تابع برداری است) نسب به مکان باشد. فرمان `help quiver(X,Y,U,V)` را در پنجره فرمان مطلب اجرا کنید:

```

QUIVER(X,Y,U,V) plots velocity vectors as arrows with components (u,v)
at the points (x,y). The matrices X,Y,U,V must all be the same size
and contain corresponding position and velocity components (X and Y
can also be vectors to specify a uniform grid). QUIVER automatically
scales the arrows to fit within the grid.

```

قبل از آنکه تابع `quiver` را بکار ببریم، لازم است یک شبکه (X, Y) از نقاط مشاهده درست کرده، و ساده ترین راه برای این منظور تابع `meshgrid` مطلب است. فرمان `meshgrid` بصورت زیر اجرا می‌شود:

```
[X,Y] = meshgrid(x,y)
```

که در آن X و Y بردارهای مقادیر تولید شبکه است. در مثال زیر ما سطح $z = -2 \leq x \leq 2$ و $-2 \leq y \leq 2$ را با قدم به اندازه یک بوسیله یک بوسیله دستور `meshgrid` شبکه‌بندی می‌کنیم. شبکه‌ای که تولید شده شامل دو آرایه مختصات ($x - y$) است.

مثال ۷.۵

```
% Example_5_7.m: example usage of meshgrid()
fprintf('This printout is from meshgrid():\n');
x = -2:1:2;
y = -2:1:2;
[X Y] = meshgrid(x,y)
%-----
```

و جواب خواهد بود:

```
This printout is from meshgrid():
X =
-2   -1    0    1    2
-2   -1    0    1    2
-2   -1    0    1    2
-2   -1    0    1    2
-2   -1    0    1    2
Y =
-2   -2   -2   -2   -2
-1   -1   -1   -1   -1
 0    0    0    0    0
 1    1    1    1    1
 2    2    2    2    2
```

بنابراین، `meshgrid` روشنی بسیار منسجم در تولید مجموعه‌های مختصات است. همان‌طوری که ملاحظه می‌کنید، تعداد عناصر در شبکه برابر حاصل‌ضرب تعداد عناصر x در تعداد عناصر y است. برای هر شبکه، یک عنصر X و یک عنصر Y وجود دارد.

مثال ۸.۵

برای نشان دادن نمودار `quiver`، می‌خواهیم میدان الکتریکی ناشی از یک بار خطی روی محور z را در صفحه $x - z$ محاسبه کنیم.

```
% Example_5_8.m
% This program plots the electric field on the X-Z plane
% for a line charge located on the Z axis and extending
% to infinity in both Z directions. The calculation is
% made over the interval xo = [-.01,.01] m and zo =
% [-.01,.01] m with a step size of 0.002 m. This example
% reuses the functions already defined in Example 5.6.
clc; clear;
```

```

global xo yo zo lambda epsilon0;
lambda=2e-9; epsilon0=8.85e-12;
% for measurements on the X-Z plane,yo will always be 0:
yo=0;
% First create a grid of X-Z locations for which
% calculate the electric field
[X,Z]=meshgrid(-10e-3:2e-3:10e-3,-10e-3:2e-3:10e-3);
% Calculate the E-field at each X-Z location, and store
% the result in the matrices Ex and Ez
for i=1:length(X(:,1))
for j=1:length(Z(1,:))
xo=X(i,j);
zo=Z(i,j);
% NOTE: don't calculate the field when situated
% ON the line charge because it will be infinity.
% Instead, use the value "NaN" (which will cause
% no point to be printed on the quiver plot).
if xo==0
Ex(i,j)=NaN;
Ez(i,j)=NaN;
else
Ex(i,j)=quad(@linecharge_Ex,-10,10);
Ez(i,j)=quad(@linecharge_Ez,-10,10);
end
end
end
% Plot the electric field
quiver(X,Z,Ex,Ez);
title('E-field due to a line charge along the Z axis');
xlabel('x axis');
ylabel('z axis');
%-----

```

شکل (۵.۵) نمودار quiver را نشان می‌دهد.

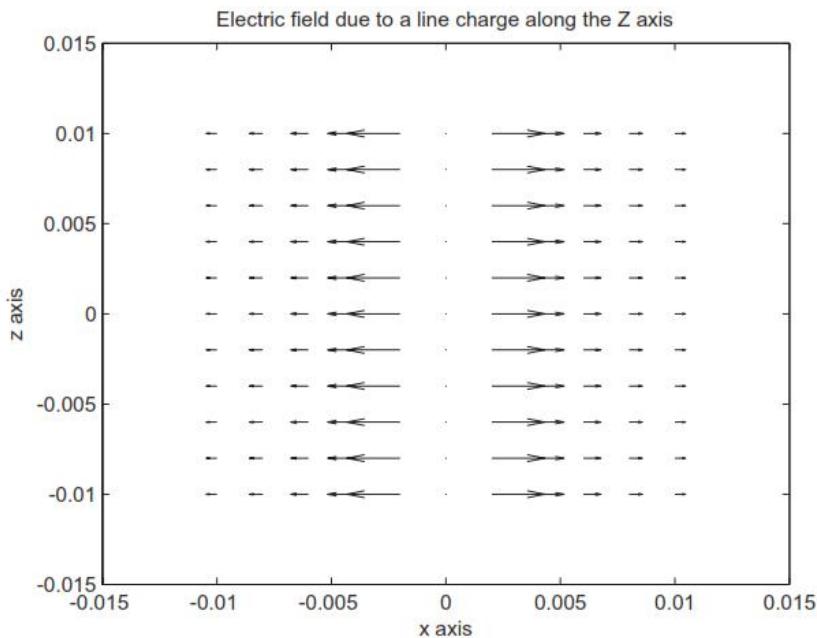
۷.۵ تابع dblquad در متلب

برای محاسبه انتگرال دوگانه متلب دارای تابع dblquad است. توضیح این تابع با نوشتن help dblquad در پنجره فرمان متلب بدست می‌آید:

```

Q = DBLQUAD(FUN,XMIN,XMAX,YMIN,YMAX) evaluates the double integral
of FUN(X,Y) over the rectangle XMIN <= X <= XMAX, YMIN <= Y <=
YMAX. FUN is a function handle. The function Z = FUN(X,Y) should
accept a vector X and a scalar Y and return a vector Z of values
of the integrand.
Non-square regions can be handled by setting the integrand

```



شکل ۵.۵: نمودار برداری میدان الکتریکی ناشی از یک خط باردار واقع در روی محور z ها در صفحه $x - z$

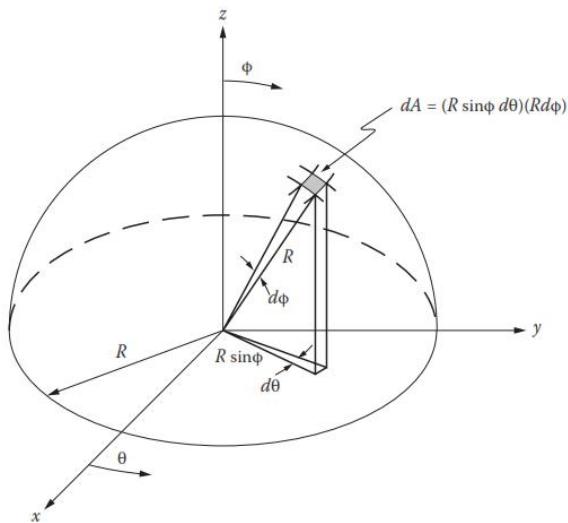
to zero outside of the region of interest. For example, the volume of a hemisphere of radius R can be determined by the dblquad function by setting $-R \leq x \leq R$ and $0 \leq y \leq R$ and setting $z = 0$ for points (x,y) that lie outside the circle of radius R around the origin.

- استفاده از dblquad شبیه quad است با این تفاوت که
- تابع FUN باید تابعی با دو متغیر (بجای یک متغیر) باشد: یک بردار مقداری (یک آرایه) در جهت x ها و یک بردار مقداری در جهت y ها.
- باید دو مجموعه کرانه وجود داشته باشد. (یکی برای x ها و یکی برای y ها).

مثال ۹.۵:

سطح و حجم یک نیم کره بشعاع یک متر را محاسبه کنید.
برای پیدا کردن سطح، ما دیفرانسیل سطح را $dA = R^2 \sin \phi d\theta d\phi$ (مطابق شکل ۶.۵) تعریف کرده و سپس انтگرال دوگانه را در کرانه های $\theta = [0, 2\pi]$ و $\phi = [0, \pi/2]$ با استفاده از dblquad گیری می کنیم. برای پیدا کردن حجم، دیفرانسیل حجم $dV = \sqrt{R^2 - x^2 - y^2} dx dy$ را مطابق شکل (۷.۵) تعریف کرده و انتگرال دوگانه را در محدوده $x = [-R, R]$ و $y = [-R, R]$ می کنیم.

```
% Example_5_9.m
% This program calculates the surface area and volume of
% a hemisphere (with radius=1) using dblquad. The exact
% values from formulae are also calculated.
```

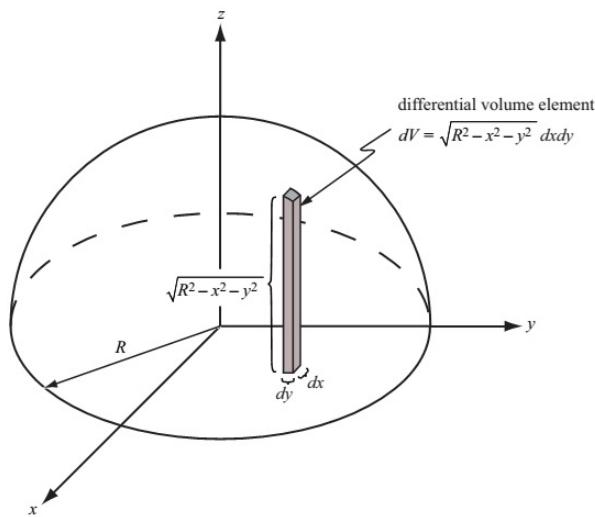


شکل ۵.۶: برای پیدا کردن سطح یک نیم‌کره، جزء سطح dA را تعریف کرده و سپس انتگرال‌گیری دوگانه را روی $d\theta$ و $d\phi$ انجام میدهیم.

```

clear; clc;
global R;
R = 1;
% calculate surface area
SA = dblquad('hemisphere_dA',0,pi/2,0,2*pi);
SA_exact = 2*pi*R^2;
% calculate volume
V = dblquad('hemisphere_dV',-R,R,-R,R);
V_exact = 2/3*pi*R^3;
% print results
fprintf('Surface area SA of a hemisphere of ');
fprintf('radius %.4f m\n',R);
fprintf('SA by DBLQUAD = %.4f m^2\n',SA);
fprintf('SA exact = %.4f m^2\n',SA_exact);
fprintf('Volume V of a hemisphere of radius %.4f m\n',R);
fprintf('V by DBLQUAD = %.4f m^3\n',V);
fprintf('V exact = %.4f m^3\n',V_exact);
%-----
% hemisphere_dA.m: This function defines the integrand
% for determining the surface area of a hemisphere and is
% used in MATLAB's dblquad function.
function [ dA ] = hemisphere_dA( phi,theta )
global R;
dA = R^2 * sin(phi);
%-----
% hemisphere_dV.m: This function defines the integrand

```



شکل ۷.۵: برای پیدا کردن حجم یک نیم‌کره، جزء حجم dV را تعریف کرده و سپس انتگرال‌گیری دوگانه را روی dx و dy انجام میدهیم.

```
% z(x,y) for determining the volume of a hemisphere and
% is used in MATLAB's dblquad function. From analytical
% geometry, the equation of a sphere is X^2 + Y^2 + Z^2 =
% R^2. Thus, dV=Z*dX*dY, where Z=sqrt(R^2-X^2-Y^2). Note
% that the hemisphere is only defined for (X,Y) points
% which lie within a circle of radius R around the
% origin, the region that is the projection of the
% hemisphere onto the X-Y plane. An 'if' statement is
% used to set Z = 0 for (X,Y) points outside the circle
% of radius R. Note: X is a vector and Y is a scalar.
function Z = hemisphere_dV(X,Y)
global R;
for i=1:length(X)
if X(i)^2 + Y^2 <= R^2
Z(i) = sqrt(R^2 - X(i)^2 - Y^2);
else
Z(i) = 0;
end
end
%-----
```

و جواب برنامه خواهد بود:

```
Surface area SA of a hemisphere of radius 1.0000 m
SA by DBLQUAD = 6.2832 m^2
SA exact      = 6.2832 m^2
Volume V of a hemisphere of radius 1.0000 m
```

V by DBLQUAD = 2.0944 m^3

V exact = 2.0944 m^3

مسائل

۱. با استفاده از بسط سری تیلور $x = 0$ حول $\log(1 + x)$ ثابت کنید که

$$\log(1 + x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \frac{1}{5}x^5 - \dots + \dots$$

۲. با استفاده از تابع quad متلب، انتگرال‌های زیر را محاسبه کنید. توجه کنید که (۵)، (۶) و (۷) انتگرال‌های نامعین هستند.

• الف :

$$I = \int_0^3 \frac{dx}{5e^{3x} + 2e^{-3x}}$$

• ب :

$$I = \int_{-\pi/2}^{\pi/2} \frac{\sin x dx}{\sqrt{1 - 4 \sin^2 x}}$$

• ج :

$$I = \int_0^\pi (\sinh x - \cos x) dx$$

• د :

$$I = \int_0^1 \frac{3e^x dx}{\sqrt{1 - x^2}}$$

• ه :

$$I = \int_0^1 \frac{\log x dx}{(1 - x)}$$

• و :

$$I = \int_0^1 \frac{\log x dx}{(1 - x^2)}$$

پروژه‌ها

۱.۵ پروژه

برای خط باردار مثال ۸.۵، میدان الکتریکی را در صفحه $y - z$ (بجای صفحه $x - z$) در فواصل $quiver$ $-50 \leq x \leq 50mm$ و $-50 \leq y \leq 50mm$ در قدمهای $10mm$ محاسبه و با استفاده از رسم کنید. توجه کنید که نباید میدان را روی خط بار دار محاسبه کنید.

۲.۵ پروژه

اکنون خط بار دار مثال ۸.۵ را بصورت محدود اصلاح می‌کنیم، بدین ترتیب که طول خط را محدود $\ell = 0.02m$ در روی محور z از $z = -0.01m$ تا $z = +0.01m$ در نظر می‌گیریم. فرض کنید $\lambda = 2 \times 10^{-9} C/m$ است.

۱. میدان الکتریکی را در فواصل $-0.05 \leq y \leq 0.05$ و $-0.05 \leq x \leq 0.05m$ در قدمهای $0.01m$ محاسبه و با استفاده از quiver رسم کنید.

۲. روی صفحه نمایشگر بازاء هر چهار مقدار x_o, z_o ، و مقادیر مربوطه E_x, E_z ، و اندازه \vec{E} را بنویسید.

۳. اکنون، فرض کنید که کل بار روی خط $Q = \lambda \ell = 0.04 \times 10^{-9} C$ بصورت بار نقطه‌ای در مبدأ مختصات قرار دارد. اندازه میدان الکتریکی را برای این نقطه باردار روی همان محدوده قبلی با استفاده از معادله (۸.۵) محاسبه کنید. توجه کنید که چون Q بار نقطه‌ای است، باید از رابطه (۸.۵) مستقیماً استفاده کنید و انتگرال گیری لازم نیست.

۴. درصد خطای را در استفاده از روش بار نقطه‌ای در محاسبه E_x و E_z در هر نقطه را بطوری که در بخش یک و سه محاسبه شد، بدست آورید. از فرمول زیر برای درصد خطای محاسباتی E_x و همین طور E_z استفاده کنید.

$$\text{خطای } E_x \% = \frac{|E_{x,1} - E_{x,2}|}{E_{x,1}}$$

که در آن

$$E_{x,1} = \text{محاسبه شده در بخش اول } E_x$$

$$E_{x,2} = \text{محاسبه شده در بخش سوم } E_x$$

در صفحه نمایشگر هر چهار مقدار x_o, z_o یکبار درصد خطای E_x و E_z را بنویسید.

۵. در چه فاصله‌ای از مبدأ مختصات بکار بردن بار نقطه‌ای خطای کمتر از ۵٪ است؟ در خیلی از حالت‌ها، اگر ما بقدر کافی از بار دور باشیم، می‌توانیم از شکل هندسی صرفنظر کرده و مسئله را با بار نقطه‌ای حل کرده و محاسبات پسیار ساده‌تر می‌شود. این را تقریب میدان دور می‌نامند.

۳.۵ پروژه

چگالی بار سطحی مثبت $\sigma = 4 \times 10^{-12} C/m^2$ تا بینهایت روی صفحه $x-y$ ، شکل (۸.۵)(الف) توزیع شده است. با استفاده از روشی که در بخش ۴.۵ توضیح داده شد، دیفرانسیل بار را $dQ = \sigma dx dy$ را تعریف کرده، و سپس میدان الکتریکی صفحه باردار را که از $[x_p = -\infty, \infty], [y_p = 0, \infty]$ هست با انتگرال گیری با مفروضات زیر محاسبه کنید.

$$E_x = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\sigma dy_p dx_p}{4\pi\epsilon_0} \frac{x_o - x_p}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}}, \quad (۱۵.۵)$$

$$E_y = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\sigma dy_p dx_p}{4\pi\epsilon_0} \frac{y_o - y_p}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}}, \quad (۱۵.۶)$$

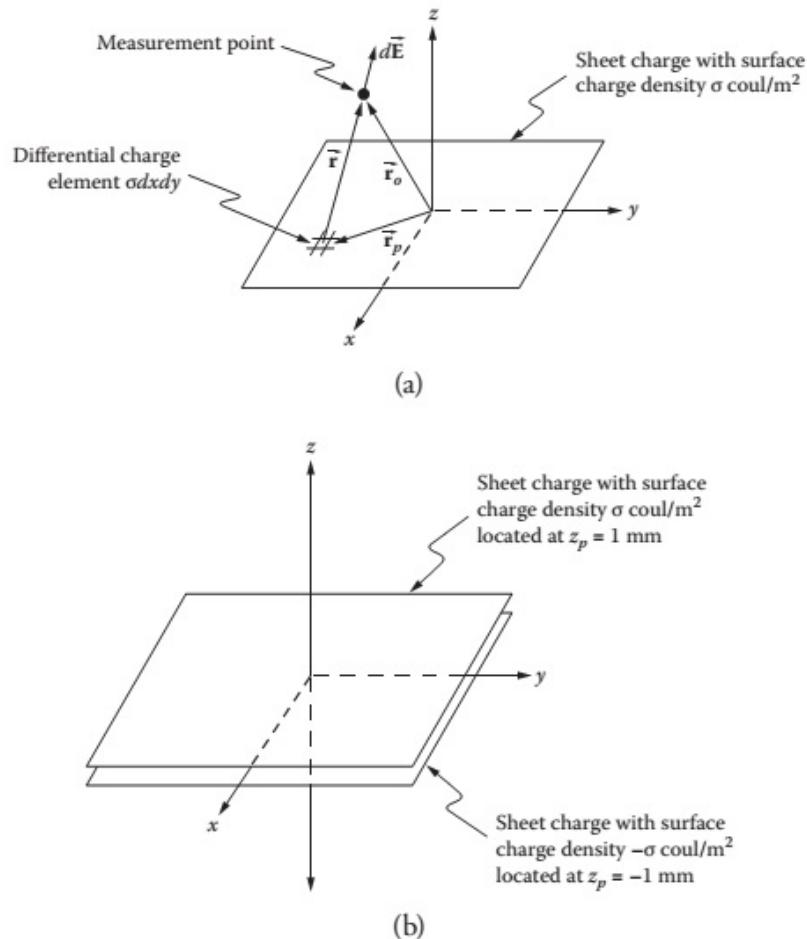
$$E_z = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\sigma dy_p dx_p}{4\pi\epsilon_0} \frac{z_o - z_p}{[(x_o - x_p)^2 + (y_o - y_p)^2 + (z_o - z_p)^2]^{3/2}}, \quad (۱۵.۷)$$

۱. برنامه‌ای بزیان متلب و استفاده از تابع dblquad نوشته و مولفه‌های E_x, E_y و E_z را در نقاط زیر محاسبه کنید.

$$x_o, y_o, z_o = [0, 0, 2 \times 10^{-3}] \quad \text{بالای صفحه}$$

$$x_o, y_o, z_o = [30 \times 10^{-3}, -5 \times 10^{-3}, 12 \times 10^{-3}] \quad \text{زیر صفحه}$$

فرض کنید ۱۰ مقدار خوبی برای تقریب بی‌نهایت است. با توجه به تقارن، مولفه‌های x و y میدان باید مقدار صفر (یا نزدیک به آن) را حاصل نمایند. این نتیجه را تائید نمائید.



شکل ۸.۵: (الف) دیفرانسیل میدان الکتریکی $\vec{d}\mathbf{E}$ در نقطه مشاهده \vec{r}_0 ناشی از المان بار $\sigma dx dy$ واقع در صفحه $z_p = -1 \text{ mm}$ و $z_p = +1 \text{ mm}$. (ب) دو صفحه موازی با بارهای مختلف العلامه واقع در $x-y$

۲. اکنون، فرض کنید که دو صفحه باردار موازی صفحه $x-y$ وجود دارد: اولی در $z_p = +1 \text{ mm}$ بوده و چگالی بار آن $\sigma = 4 \times 10^{-12} \text{ C/m}^2$ و صفحه دوم در $z_p = -1 \text{ mm}$ و چگالی بار آن $\sigma = -4 \times 10^{-12} \text{ C/m}^2$ است (به شکل ۸.۵(ب) مراجعه کنید). میدان الکتریکی منتجه را محاسبه میدان یکی هریک از صفحات و سپس جمع کردن آنها بدست آورید. مولفه‌های E_x , E_y , و E_z را در نقاط زیر محاسبه کنید.

$$x_o, y_o, z_o = [0, 0, 2 \times 10^{-3}] \quad \text{بالای صفحات}$$

$$x_o, y_o, z_o = [10 \times 10^{-3}, 0, -0.5 \times 10^{-3}] \quad \text{بین صفحات}$$

$$x_o, y_o, z_o = [20 \times 10^{-3}, -5 \times 10^{-3}, 12 \times 10^{-3}] \quad \text{زیر صفحات}$$

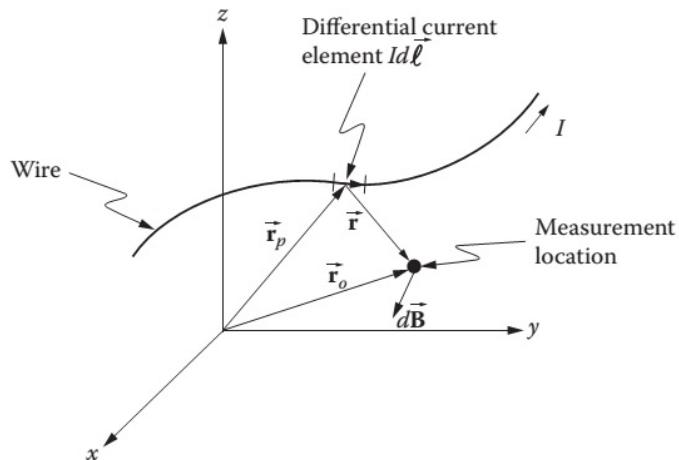
نتایج را چاپ کنید. آیا نتایج معقول است؟

قانون بیو-ساوار جریان الکتریکی را به میدان مغناطیسی مربوط کرده و آنرا برای سیم‌ها بصورت زیر بیان میدارد (به شکل ... مراجعه کنید).

$$d\vec{B} = \frac{\mu_0}{4\pi} \frac{Id\vec{l} \times \hat{e}_r}{|\vec{r}|^2} \quad (16.5)$$

که در آن $d\vec{B}$ بردار میدان مغناطیسی جزئی، μ_0 نفوذپذیری فضای آزاد و برابر $4\pi \times 10^{-7}$ هانری بر متر، I جریان بر حسب آمپر، $d\vec{l}$ تکه طول سیم همراه جهت جریان مربوطه بر حسب آمپر، \vec{r} برداری از عنصر جریان به نقطه اندازه‌گیری، \hat{e}_r بردار واحد در جهت \vec{r} و \times نمایشگر ضرب برداری است. در شکل ... ما نیز \vec{r}_p را عنوان محل عنصر جریان، و \vec{r}_o را عنوان محل اندازه‌گیری بطوری که $\vec{r} = \vec{r}_o - \vec{r}_p$ است انتخاب کردیم

می‌خواهیم میدان مغناطیسی یک سیم بی‌نهایت طویل را که روی محور z ها واقع بوده و جریان $I = 1 \text{ mA}$ را در جهت $+z$ حمل می‌کند محاسبه کنیم. توجه کنید که، در واقعیت، یک سیم طویل بی‌نهایت حامل جریان وجود ندارد زیرا اصل بقاء بار الکتریکی را نقض می‌کند. اکنون در اینجا لازم است فرض کنیم سیم در $\pm\infty$ به منبع جریان وصل شده است.



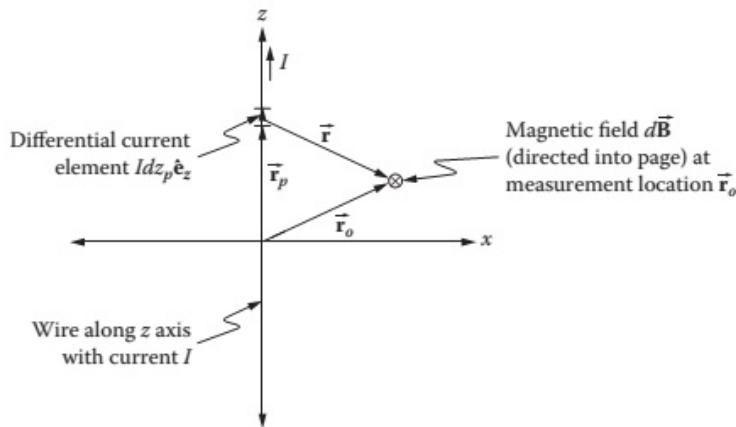
شکل ۹.۵: دیفرانسیل میدان مغناطیسی $d\vec{B}$ در نقطه مشاهده \vec{r}_o ناشی از یک المان جریان $Id\vec{l}$

۱. بطوری که در شکل (۱۰.۵) نشان داده شده است، برای سیم در جهت محور z ها، $d\vec{l} = dz\hat{e}_z$ است. بعلت ضرب خارجی در معادله (۱۶.۵)، اگر $d\vec{l}$ در جهت z هابشد، در این صورت میدان $d\vec{B}$ حاصله فقط مولفه‌هایی در جهت x و y دارد. دو معادله جدا از یکدیگر برای dB_x و dB_y با نوشتن \hat{e}_r بر حسب مولفه‌های \hat{e}_x , \hat{e}_y , \hat{e}_z و \hat{e}_r نوشته، و ضرب خارجی بین $d\vec{l}$ و \hat{e}_r را محاسبه کرده و سپس مولفه‌های x و y میدان $d\vec{B}$ را جدا کنید. راهنمائی: \hat{e}_x و \hat{e}_y را از \hat{e}_r با استفاده از روشی که در معادله (۱۰.۵) بکار برده شد استفاده کنید.

۲. دوتابع متلب برای dB_x و dB_y که میدان مغناطیسی را در جهت‌های x و y بر حسب تک آرگومان z_p محاسبه کند بنویسید. از متغیر global برای اندازه‌گیری موقعیت x_o , y_o , z_o استفاده کنید. ازتابع quad برای انتگرال گیری در بازه $z_p = [-\infty, \infty]$ استفاده کرده و میدان مغناطیسی را در نقاط زیر محاسبه کنید:

$$x, y, z = [1 \times 10^{-4}, 1 \times 10^4, 0]$$

$$x, y, z = [-1 \times 10^{-4}, 1 \times 10^4, 0]$$



شکل ۱۰.۵: میدان مغناطیسی ناشی از یک سیم طویل که در روی محور z قرار دارد و فقط مولفه‌هایی در جهت x و y دارد.

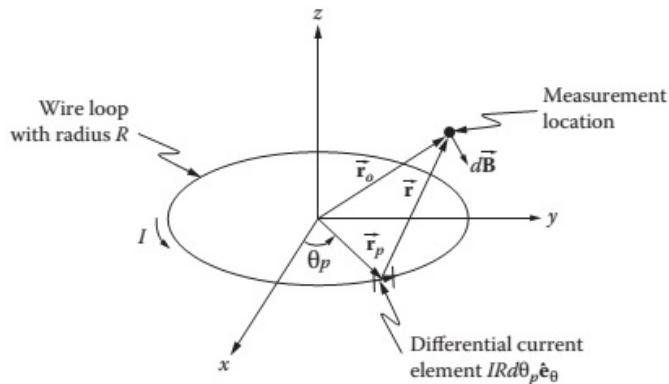
$$x, y, z = [-1 \times 10^{-4}, -1 \times 10^4, 0]$$

$$x, y, z = [1 \times 10^{-4}, -1 \times 10^4, 0]$$

فرض کنید ۱۰ تقریب کافی برای بینهایت است. جواب‌ها را بنویسید. در این راه حل در مورد تقارن بحث کنید.

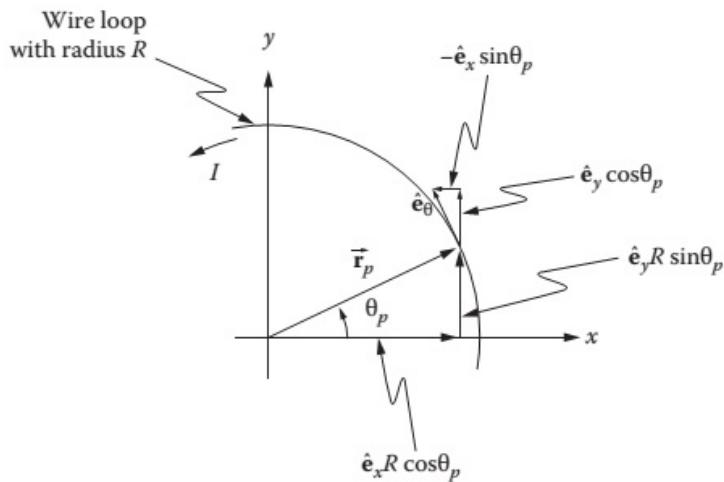
۳. ازتابع meshgrid استفاده کرده و مجموعه‌ای از نقاط برای \vec{r}_o بدست آورده که در آن $-200 \leq x_o \leq 200$ mm، $-200 \leq y_o \leq 200$ mm، $z_o = 0$ باشد. برای x_o و y_o و z_o قدمها را ۵۰ mm در نظر گیرید. میدانهای مغناطیسی B_x و B_y را در بازاء هر مقدار \vec{r}_o محاسبه کرده و آنرا با quiver رسم کنید. وقتی در محاسبات مقادیر \vec{r}_o تکرار می‌کنید، مطمئن باشید که نقطه مبداء را درنظر نگیرید.

• پروژه ۵.۵
شکل (۱۱.۵) یک حلقه سیمی به شعاع R که مرکز آن واقع در مبدأ مختصات بوده و جریان I



شکل ۱۱.۵: دیفرانسیل المان میدان مغناطیسی $d\vec{B}$ در نقطه مشاهده \vec{r}_o ناشی از یک حلقه سیمی جریاندار واقع در صفحه $x-y$

را حمل می‌کند نشان میدهد. با استفاده از قانون بیو-ساوار معادله (۱۶.۵)، می‌توانیم دیفرانسیل میدان مغناطیسی را بر حسب دیفرانسیل المان جریان در آن $Rd\theta_P Id\vec{l} = IRd\theta_P \hat{\mathbf{e}}_\theta$ که در آن دیفرانسیل طول در مختصات استوانه‌ای المان جریان واقع در \vec{r}_p و $\hat{\mathbf{e}}_\theta$ بردار واحد در جهت θ_P است. در این نوع مسائل با تقارن استوانه‌ای، راه حل ساده این است که بردار واحد استوانه‌ای را به بردارهای واحد کارتزین تبدیل کنیم، اما هنوز هم انتگرال‌گیری با استفاده از متغیرهای استوانه‌ای انجام می‌شود. شکل (۱۲.۵) نشان میدهد که چگونه \vec{r}_p و $\hat{\mathbf{e}}_\theta$ را می‌توان بر حسب مولفه‌های x و y با استفاده از بردارهای واحد $\hat{\mathbf{e}}_x$ و $\hat{\mathbf{e}}_y$ بیان کرد:



شکل ۱۲.۵: بردارهای \vec{r}_p و $\hat{\mathbf{e}}_\theta$ را می‌توان بر حسب مولفه‌های x و y برای حل میدان مغناطیسی توسط تابع quad مطلب تجزیه کرد.

$$\vec{r}_p = \hat{\mathbf{e}}_x R \cos \theta_p + \hat{\mathbf{e}}_y R \sin \theta_p \quad (17.5)$$

$$\hat{\mathbf{e}}_\theta = -\hat{\mathbf{e}}_x \sin \theta_p + \hat{\mathbf{e}}_y \cos \theta_p \quad (18.5)$$

اکنون می‌توانیم معادله (۱۶.۵) را بصورت زیر بازنویسی کنیم

$$d\vec{\mathbf{B}} = \frac{\mu_0 IR}{4\pi} d\theta_p \frac{(-\hat{\mathbf{e}}_x \sin \theta_p + \hat{\mathbf{e}}_y \cos \theta_p) \times (\hat{\mathbf{e}}_x(x_o - R \cos \theta_p) + \hat{\mathbf{e}}_y(y_o - R \sin \theta_p) + \hat{\mathbf{e}}_z(z_o - z_p))}{[(x_o - R \cos \theta_p)^2 + (y_o - R \sin \theta_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (19.5)$$

که در آن $\mu_0 = 4\pi \times 10^{-7} H/m$ نفوذپذیری مغناطیسی^۴ فضای آزاد است.

۱. حاصل ضرب بردار خارجی در معادله (۱۹.۵) را محاسبه کرده و جواب حاصل را به صورت سه معادله برای مولفه‌های x , y و z دیفرانسیل میدان مغناطیسی بنویسید. با استفاده از اینها برای dB_z , dB_x و dB_y برنامه‌ای بزبان متلب نوشته که آرگومان (θ_p) را گرفته و با استفاده از تابع quad متلب B_z , B_y و B_x را محاسبه نماید.

۲. با استفاده از تابع meshgrid مجموعه‌ای از نقاط در صفحه $-8 \leq x_0 \leq 8$ و $-2 \leq z_0 \leq 2$ mm با قدمهای ۱ mm و ۰.۵ mm تولید کنید. در هر نقطه این مجموعه x و z را محاسبه کنید. به بازه انتگرال توجه بیشتری مبذول دارید.

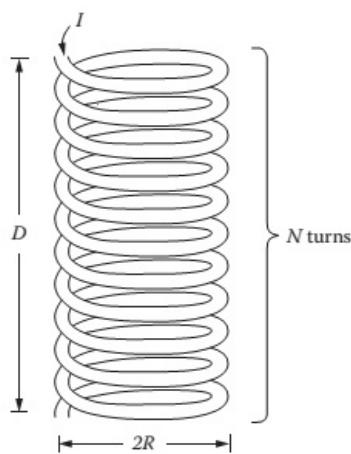
⁴permeability

ضمناً با استفاده از quiver میدان مغناطیسی را رسم کنید. فرض کنید $R = 2.5 \text{ mm}$ و $I = 1 \text{ mA}$.

۶.۵ پروژه

یک سلونوئید یک سیم مارپیچ است که برای تولید میدان مغناطیسی برای تحریک محرکها بکار می‌رود. یکی از آنها رله‌های الکتریکی است که شیرها آب الکتریکی، دندنه استارت خودروها بکار می‌رود. شکل (۱۳.۵) یک سلونوئید بشعاع R و طول D با N دور سیم‌پیچ و جریان I را نشان می‌دهد.

ما این سلونوئید را در مطلب بصورت یک سطح استوانه‌ای جریان با چگالی $\vec{K} = \frac{NI}{D}\hat{e}_\theta A/m$ که چگالی جریان در جهت θ است، مدل می‌کنیم. قانون بیو-ساوار برای جریان سطحی برابر است با:



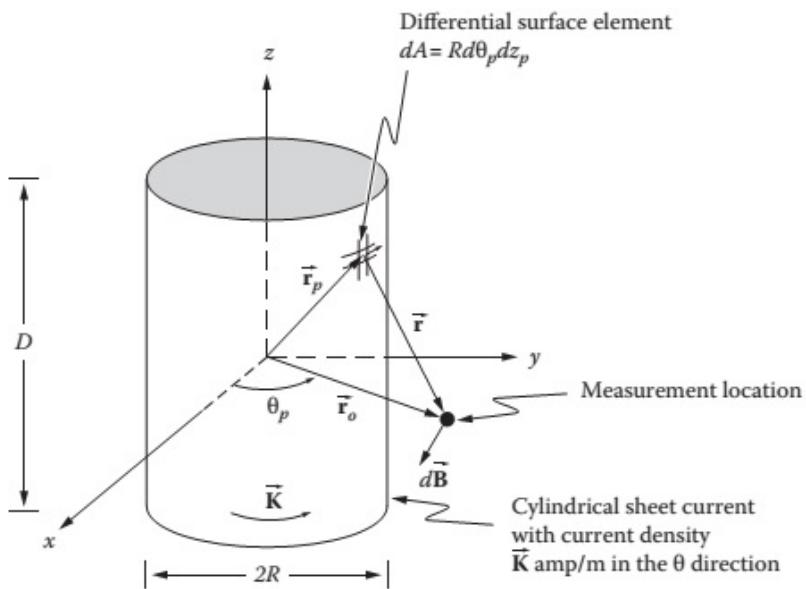
شکل ۱۳.۵: یک سلونوئید شامل N دور سیم، بشعاع R ، بطول D ، و بجریان I

$$d\vec{B} = \frac{\mu_0}{4\pi} \frac{\vec{K} dA \times \hat{e}_r}{|\vec{r}|^2} \quad (20.5)$$

که در آن dA دیفرانسیل دو بعدی المان سطح، \vec{r} برداری از دیفرانسیل سطح به نقطه مشاهده است، \hat{e}_r بردار واحد در جهت \vec{r} ، \times علامت ضرب خارجی، و $\mu_0 = 4\pi \times 10^{-7} \text{ H/m}$ نفوذپذیری مغناطیسی برای فضای آزاد است. از شکل (۱۴.۵) مشاهده می‌کنیم که در مختصات استوانه‌ای، $dA = Rd\theta_p dz$ ، و \vec{r}_p محل المان سطح (در مختصات استوانه‌ای دارای مولفه‌های r_p ، θ_p ، z_p) و \hat{e}_θ واحد در جهت θ_p است. ضمناً نقطه مشاهده را $\vec{r}_o = x_o\hat{e}_x + y_o\hat{e}_y + z_o\hat{e}_z$ تعریف می‌کنیم. از هندسه مسئله می‌توان $\vec{K} dA$ نوشت و \vec{r} بر حسب بردارهای واحد کارتزین برابر است با:

$$d\vec{K} dA = \frac{NI}{D} Rd\theta_p dz_p (-\hat{e}_x \sin \theta_p + \hat{e}_y \cos \theta_p) \quad (21.5)$$

$$\vec{r} = \vec{r}_o - \vec{r}_p = \hat{e}_x(x_o - R \cos \theta_p) + \hat{e}_y(y_o - R \sin \theta_p) + \hat{e}_z(z_o - z_p) \quad (22.5)$$



شکل ۱۴.۵: یک سلونوئید را می‌توان با جریان سطحی $\vec{K} A/m$ مدل کرد.

با قرار دادن در رابطه (۲۰.۵) خواهیم داشت:

$$d\vec{B} = \frac{\mu_0 N I R}{4\pi D} d\theta_p dz_p \frac{(-\hat{e}_x \sin \theta_p + \hat{e}_y \cos \theta_p) \times [\hat{e}_x(x_o - R \cos \theta_p) + \hat{e}_y(y_o - R \sin \theta_p) + \hat{e}_z(z_o - z_p)]}{[(x_o - R \cos \theta_p)^2 + (y_o - R \sin \theta_p)^2 + (z_o - z_p)^2]^{3/2}} \quad (۲۳.۵)$$

۱. حاصل ضرب خارجی را در رابطه (۲۳.۵) محاسبه کرده و نتیجه را در سه معادله جداگانه برای مولفه‌های x , y و z دیفرانسیل میدان مغناطیسی بنویسید. برای برنامه‌ای بزبان متلب نوشته که آرگومان p (θ_p) و z_p را گرفته و دیفرانسیل میدان B را بترتیب در جهت‌های x , y و z محاسبه نماید.

۲. فرض کنید مرکز سلونوئید در مبدأ مختصات قرار داشته و شعاع آن $R = 2.5 mm$ و طول آن $D = 15 mm$ باشد. با استفاده از تابع `meshgrid` مجموعه‌ای از نقاط $[X, Y, Z]$ سه‌بعدی نقاط اندازه‌گیری برای \vec{r} تولید کنید. توجه کنید که `meshgrid` آرگومان سومی برای مجموعه سه بعدی می‌تواند در نظر گیرد. (با نوشتن `help meshgrid` در باره این موضوع مفصلًاً مطالعه نمائید). با استفاده از بازه‌های (x_o, y_o, z_o) در نظر گیرید. (با نوشتن `help dblquad` در $-3 \leq x_o \leq 3 mm$, $0 \leq y \leq 3 mm$, $-10 \leq z_o \leq 10 mm$ و $1.5 mm \leq R \leq 5 mm$ مجموعه سه بعدی نقاط را درست کنید).

۳. از تابع `dblquad` استفاده کرده و B_z , B_y , B_x را در هر نقطه از مجموعه محاسبه کنید. (محدوده انتگرال گیری را درست در نظر گیرید). فرض کنید که $N = 500$ و $I = 1 mA$ است.

۴. یک ترسیم سه بعدی `quiver` با استفاده از تابع `quiver3` مطلب رسم کنید. برای این کار با نوشتن `help quiver3` اطلاعات مفصلی کسب کنید.

فصل ۶

انتگرال‌گیری عددی معادلات دیفرانسیل معمولی

۱.۶ مقدمه

در این فصل، ما چندین روش را برای حل معادلات دیفرانسیل معمولی^۱ مطالعه خواهیم کرد. معادلات دیفرانسیل معمولی یا ODE را می‌توان به دو دسته تقسیم نمود:

• **الف:** مسائل مقدار اولیه که در آن شرایط اولیه لازم را میدانیم. مثلاً، مقدار در یک مدر و لتأثر گره‌ها (یا مشتق آنها) در لحظه $t = 0$ معلوم است.

• **ب:** مسائل مقدار مرزی که در آنها مقادیر متغیرهای مستقل در مختصات مشخصی در هندسه مسئله معلوم است. مثلاً، پتانسیل الکتریکی در هر دو انتهای یک هادی معلوم است.

برای مسائل مقدار اولیه، ما چندین روش انتگرال‌گیری عددی، شامل روش اولر^۲، روش اولر اصلاح شده، روش رانج کوتا^۳ و تابع `ode45` مطلب است. برای مسائل مقادیر مرزی، ما از روش تفاضلات محدود^۴ که معادلات دیفرانسیل معمولی را بیک سامانه معادلات خطی جبری تقلیل داده و آن را می‌توان با روش معادلات جبری سه قطعی حل کرد.

۲.۶ مسئله مقدار اولیه

در یک مسئله مقدار اولیه، مقادیر متغیر مستقل و مشتقهای لازم در نقطه‌ای که انتگرال‌گیری شروع می‌شود معلوم است. ما با یک معادله دیفرانسیل مرتبه اول بفرم کلی بنحوی که مشتق بر حسب تابعی از x و y و شرایط اولیه $y(0) = Y_I$ معلوم است، شروع می‌کنیم:

$$\frac{dy}{dx} = f(x, y), \quad y(0) = Y_I \quad (1.6)$$

¹ordinary differential equations (ODE)

²Euler method

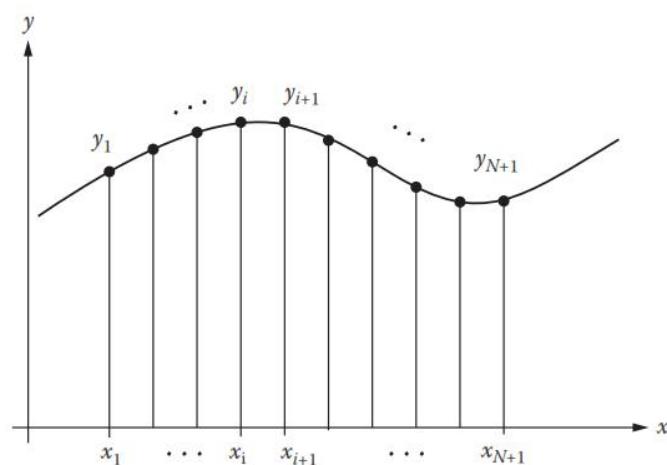
³Runge-Kutta

⁴finite difference methods

چندین روش برای حل این نوع مسئله وجود دارد، که شامل روش اولر است که ساده بوده و خیلی کم بکار برده می‌شود؛ روش اولر اصلاح شده؛ روش رانج-کوتا؛ و سایر روشها. اینگونه روشها دارای افت و خیزهایی نسبت به ساده‌بودن، دقت، و بازدهی محاسباتی دارد.

۳.۶ آلگوریتم اولر

رویکرد کلی برای حل معادلات دیفرانسیل بصورت عددی تقسیم حوزه x به N قسمت $x_1, x_2, x_3, \dots, x_{N+1}$ و سپس "پیشروی"^۴ در جهت x در این فواصل درحالی که $y_2, y_3, y_4, \dots, y_{N+1}$ را محاسبه می‌نماییم. توجه کنید که y_1 مشخص است و برابر شرایط اولیه Y_I می‌باشد. به شکل (۱.۶) مراجعه کنید. بسط



شکل ۱.۶ :

سری تیلور^۵ حول نقطه دلخواه x_i می‌دهد

$$y(x) = y(x_i) + \frac{y(x_i)}{1!}(x - x_i) + \frac{y(x_i)}{2!}(x - x_i)^2 + \dots \quad (۲.۶)$$

اگر فقط اولین دو جمله سری را در نظر گیریم، می‌توانیم مقدار تقریبی $y(x_{i+1})$ (که بعداً با y_{i+1} نشان خواهیم داد) بصورت زیر داشته باشیم:

$$\begin{aligned} y_{i+1} &\approx y(x_i) + \frac{y(x_i)}{1!}(x_{i+1} - x_i) \\ &\approx y_i + y_i b \end{aligned} \quad (۳.۶)$$

و در آن $b = x_{i+1} - x_i$ که بنام **طول قدم**^۴ تعریف می‌شود. و y_i شب منحنی $y(x)$ در x_i است. با قراردادن معادله (۱.۶) در (۳.۶) خواهیم داشت:

$$y_{i+1} = y_i + bf(x_i, y_i) \quad (۴.۶)$$

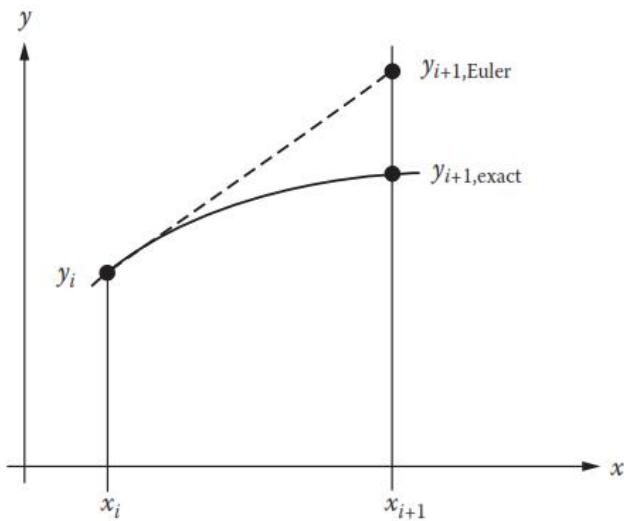
بطوری که از شکل (۲.۶) مشاهده می‌شود، پیشگوئی y_{i+1} با روش اولر از مقدار واقعی y_{i+1} بالا می‌زند.

$$y_2 = y_1 + bf(x_1, y_1) = Y_I + bf(x_1, Y_I)$$

^۴march

^۵Taylor series

\step size



شکل ۲.۶: در روش اولر، ابتداء از تقریب سری مرتبه اول تیلور (یعنی خط مستقیم) استفاده کرده تا مقدار بعدی y را محاسبه کنیم.

بهمنین منوال، y_3 را می‌توان با قرار دادن $i = 2$ در رابطه (۴.۶) بدست آورد:

$$y_3 = y_2 + b f(x_2, y_2)$$

فرآیند ادامه یافته تا مقادیر $y_4, y_5, y_6, \dots, y_{N+1}$ بدست آید. روش اولر یک نمونه از روش صریح^{۳.۶} است زیرا هر مقدار جدید y_{i+1} صرفاً از مقادیر قبلی x و y بدست می‌آید.

۴.۶ روش اولر اصلاح شده

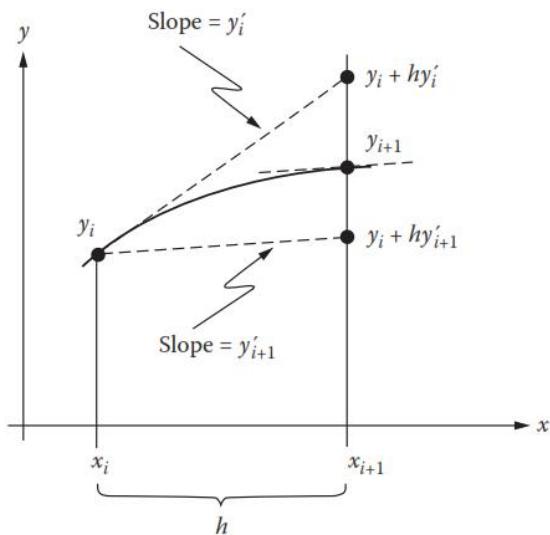
روش اولر اصلاح شده دقت بیشتری برای محاسبات مقدار y_{i+1} نسبت به تقریب روش اولر خطی ساده که در بخش قبل بحث شد دارد. دوباره، معادله دیفرانسیل فرم استاندارد $y' = f(x, y)$ را با مقدار اولیه $y(0) = Y_I$ را در نظر گرفته و دوباره حوزه x را به N قسمت تقسیم و در جهت x پیش روی می‌کنیم. مشاهده نمودید که در روش اولر، با گرفتن $y_{i+1} = y_i + b f(x_i, y_i)$ بالازدن از مقدار واقعی y_{i+1} خواهیم داشت. شکل (۲.۶) را مشاهده کنید. اکنون، فرض کنید که می‌توانیم $y(x_{i+1})$ را تعیین نمائیم که همان شبیه منحنی در x_{i+1} است. اگر در معادله (۳.۶)، y_{i+1} را با استفاده از $y(x_{i+1})$ بتوانیم پیشگوئی کنیم، یعنی

$$y_{i+1} = y_i + b y_{i+1} = y_i + b f(x_{i+1}, y_{i+1}) \quad (5.6)$$

در آن صورت می‌توانیم از بالازدن از مقدار واقعی y_{i+1} جلوگیری کرده و آنرا بزرگ‌نماییم. این در شکل (۳.۶) نشان داده شده است. در اینجا، مماس بر منحنی را در y_{i+1} ساخته و خطی موازی که از نقطه (x_i, y_i) بگذرد رسم نموده تا مقدار پیشگوئی y_{i+1} را تعیین کنیم. چون استفاده از y_i در معادله (۳.۶) از مقدار واقعی y_{i+1} بالامیزند و استفاده از y_{i+1} در معادله (۳.۶) از مقدار واقعی پائین‌تر می‌آورد، بهتر است که از میانگین این دو مشتق در معادله (۳.۶) استفاده کنیم، یعنی،

$$y_{i+1} = y_i + b \frac{y_i + y_{i+1}}{2} \quad (6.6)$$

^{۳.۶}explicit



شکل ۳.۶: در روش اولر اصلاح شده، ما از تقریب‌های میانگین دو خط مستقیم برای پیشگوئی y_{i+1} استفاده می‌کنیم.

متاسفانه، معادله (۶.۶) دیگر صریح نیست زیرا ما مقدار y_{i+1} را نمی‌دانیم. استفاده از معادله (۶.۶) در حل معادله دیفرانسیل (۱.۶) نمونه‌ای از یک روش ضمنی^۹ است. البته، می‌توانیم یک مقدار برای y_{i+1} با استفاده از روش تصحیح پیشگوئی^{۱۰} تقریب زنیم. برای اعمال این روش، معادله (۶.۶) را بصورت زیر بازنویسی می‌کنیم:

$$y_{i+1}^C = y_i + b \frac{y_i + (y_{i+1})^P}{2} \quad (7.6)$$

که در آن اندیس P نمایشگر مقدار پیشگوئی^{۱۱}، و اندیس C نمایشگر مقدار تصحیح^{۱۲} است. معادله (۷.۶) را معادله تصحیح شده می‌نامند و می‌توان در محاسبه مقدار y_{i+1} بصورت تکراری استفاده کرد. با قرار دادن معادله (۱.۶) در (۷.۶) خواهیم داشت

$$y_{i+1}^C = y_i + \frac{b}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1}^P)] \quad (8.6)$$

مراحل روش تصحیح پیشگوئی بقرار زیر است:

۱. از روش اولر برای تعیین اولین مقدار پیشگوئی برای y_{i+1} استفاده کنید و آنرا $y_{i+1}^{P_1}$ نام‌گذاری می‌کنیم، یعنی

$$y_{i+1}^{P_1} = y_i + bf(x_i, y_i)$$

۲. ابتدا مقدار تصحیح شده از $y_{i+1}^{P_1}$ با استفاده از $y_{i+1}^{C_1}$ در معادله (۸.۶) محاسبه کنید.

۳. از $y_{i+1}^{C_1}$ بعنوان مقدار پیشگوئی شده جدید $y_{i+1}^{P_2}$ در معادله (۸.۶) استفاده کنید.

۴. مقدار تصحیح شده جدید $y_{i+1}^{C_2}$ را محاسبه نمایید.

^۹implicit

^{۱۰}predictor-corrector

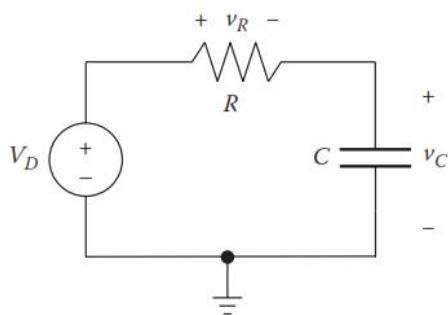
^{۱۱}predicted value

^{۱۲}corrected value

۵. مراحل سه و چهار را تکرار کرده تا $\epsilon = |y_{i+1}^{C_{n+1}} - y_{i+1}^{C_n}|$ شود. در اینجا ϵ تولرانس خطای است که بستگی به دقیقیت مورد نظر شما دارد و معمولاً کسری از درصد مقدار تصحیح شده است، برای مثال $\epsilon = 0.01\% \times y_{i+1}^{C_n}$.

مثال ۱.۶:

در این مثال، میخواهیم معادله دیفرانسیل مرتبه اولی را دقیق حل کرده و سپس آنرا با جواب روش اولر و روش اولر اصلاح شده مربوطه مقایسه کنیم.



شکل ۴.۶: مدار RC

یک مدار RC ساده با ولتاژ منبع V_D مانند شکل (۴.۶) تغذیه می‌شود. با نوشتتن قانون ولتاژ کیرشهوف در یک حلقه بسته خواهیم داشت:

$$V_D - v_R - v_C = 0 \quad (9.6)$$

قانون اهم را ($v_R = iR$) اعمال کرده و رابطه ولتاژ و جریان خازن $i = C \frac{dv_C}{dt}$ را جایگزین می‌نمائیم، بنابراین بازنویسی معادله (۹.۶) خواهد شد

$$V_D - RC \frac{dv_C}{dt} - v_C = 0$$

و آنرا مرتب کرده

$$\frac{dv_C}{dt} + \frac{1}{\tau} v_C = \frac{V_D}{\tau} \quad (10.6)$$

که در آن $\tau = RC$ ثابت زمانی^{۱۳} مدار RC است. معادله دیفرانسیل مرتبه اول از v_C نسبت به زمان است. فرض می‌کنیم که در $t = 0$ ولتاژ اولیه خازن صفر، و ولتاژ منبع تغذیه یکتابع تک شیبی^{۱۴}، یعنی $v_D(t) = V_0 t$ باشد. ابتداء معادله (۱۰.۶) را بصورت دقیق حل کرده و بعداً بعنوان مبنای مقایسه با جواب عددی می‌توانیم از آن استفاده کنیم. ما با حدس جواب خصوصی $v_{C,P}$ برابر باتابع تغذیه مدار کار را شروع می‌کنیم:

$$v_{C,P} = \alpha t + \beta \quad (11.6)$$

که در آن α و β مقادیر ثابتی هستند که باید تعیین شوند. با قرار دادن $v_{C,P}$ در معادله (۱۰.۶) خواهیم داشت

$$\frac{d}{dt}(\alpha t + \beta) + \frac{1}{\tau}(\alpha t + \beta) = \frac{V_0 t}{\tau}$$

که بما میدهد

$$\frac{\alpha t}{\tau} + \alpha + \frac{\beta}{\tau} = \frac{V_0 t}{\tau} \quad (12.6)$$

^{۱۳}time constant

^{۱۴}ramp function

با تطبیق جملات دوطرف معادله (۱۲.۶) مشاهده می‌شود که $\alpha = V_0 \tau$, $\beta = -V_0 \tau$. بنابراین،
 $v_{C,P} = V_0 t - V_0 \tau$

بعداً، جواب مکمل $v_{C,H}$ معادله همگن، با صفر قرار دادن تابع تغذیه در معادله (۱۰.۶)، پیدا می‌کنیم.

$$\frac{dv_{C,H}}{dt} + \frac{1}{\tau} v_{C,H} = 0 \quad (13.6)$$

معادله (۱۳.۶) قابل جدا سازی است. بنابراین،

$$\frac{dv_{C,H}}{v_{C,H}} = \frac{dt}{\tau} \quad \rightarrow \quad \ln v_{C,H} = -\frac{t}{\tau} + \ln \gamma \quad \rightarrow \quad v_{C,H} = \gamma e^{-t/\tau} \quad (14.6)$$

که در آن γ مقداری ثابتی است که باید تعیین شود. اکنون جواب کامل را بصورت جمع جواب خصوصی و جواب همگن در نظر می‌گیریم:

$$\begin{aligned} v_C &= v_{C,P} + v_{C,H} \\ &= V_0 t - V_0 \tau + \gamma e^{-t/\tau} \end{aligned} \quad (15.6)$$

شرایط اولیه $v_C(0) = 0$ را در معادله (۱۵.۶) اعمال کرده و $\gamma = V_0 t$ بدست می‌آید. بنابراین، جواب کامل برای پاسخ تابع تک شیبی مدار RC برابر است با:

$$v_C(t) = V_0 \left[t - \tau(1 - e^{-t/\tau}) \right] \quad (16.6)$$

اکنون معادله (۱۰.۶) را برای پارامترهای $V_0 = 10 V$, $C = 1 \mu F$, $R = 1 k\Omega$ بروش عددی حل می‌کنیم. اول، معادله دیفرانسیل (۱۰.۶) بصورت معادله (۱.۶) تبدیل می‌کنیم:

$$\frac{dv_C}{dt} = \frac{1}{\tau}(V_D - v_C) = \frac{1}{\tau}(V_0 t - v_C) = f(t, v_C) \quad (17.6)$$

بعداً، ما بازه زمانی و قدم‌های زمانی را انتخاب می‌کنیم. برای مدار RC، میدانیم که رفتار گذرا مطلوب ما در چند برابر ثابت زمانی مدار اتفاق می‌افتد، و ازاین‌رو ما بازه زمانی را $t = [0, 10\tau]$ انتخاب می‌کنیم.

ضمناً این بازه زمانی را به بیست قدم تقسیم می‌کنیم، لذا $h = \frac{10\tau}{20}$ است.

متن برنامه بزبان متلب روش اولر و روش اولر اصلاح شده را بصورت زیر بیان میدارد:

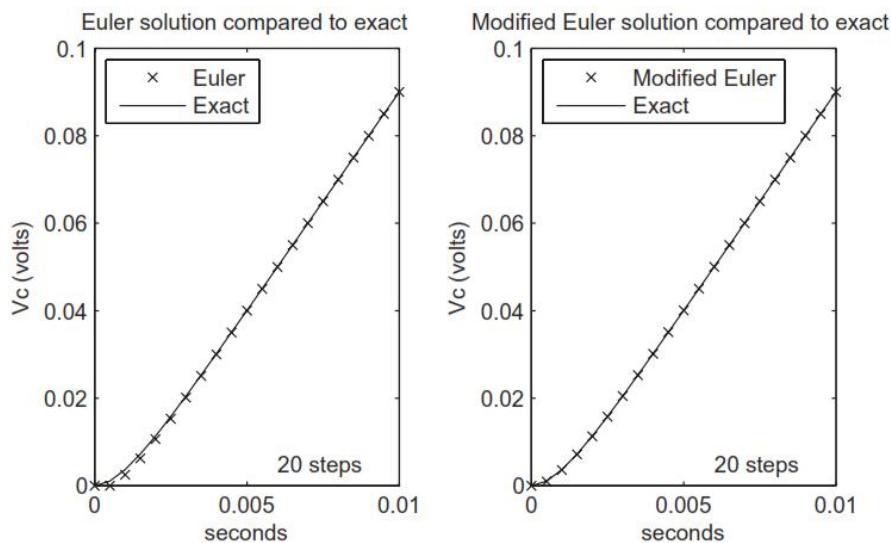
```
% Example_6_1.m
% Calculate ramp response of RC circuit via 3 methods:
% 1. exact solution
% 2. by Euler method
% 3. by modified Euler method
clc; clear;
% Circuit parameters:
R=1e3; % ohms
C=1e-6; % farads
tau = R*C; % seconds
Vo = 10; % volts
f = @(tm,Vc) (1/tau)*(Vo*tm-Vc); % Equation 6.17
% Parameters for Euler algorithm
interval = 10*tau;
steps = 20;
h = interval/steps;
% Create set of time points to calculate
t = 0:h:interval;
```

```
% Initial conditions
Vc_exact(1)=0;
Vc_euler(1)=0;
Vc_mod_euler(1)=0;
% Exact and Euler method: march in time
for i=1:length(t)-1
% Apply Equation 6.16
Vc_exact(i+1) = Vo*(t(i+1)-tau+tau*exp(-t(i+1)/tau));
% Apply Equation 6.4
Vc_euler(i+1) = Vc_euler(i) + h*f(t(i),Vc_euler(i));
end
% Modified Euler: apply corrector equation
tolerance = 1e-4; % .01 percent
for i=1:length(t)-1
% Apply Equation 6.16
Vc_exact(i+1) = Vo*(t(i+1)-tau+tau*exp(-t(i+1)/tau));
% Apply Equation 6.4
Vc_euler(i+1) = Vc_euler(i) + h*f(t(i),Vc_euler(i));
end
% Modified Euler: apply corrector equation
tolerance = 1e-4; % .01 percent
for i=1:length(t)-1
% use the Euler method to calculate an initial guess
% for the predicted value
Vc_predicted = Vc_mod_euler(i) ...
+ h*f(t(i),Vc_mod_euler(i));
% Repeatedly apply the corrector equation until the
% predictor and corrector are equal (or within
% tolerance)
for j=1:50
Vc_corrected = Vc_mod_euler(i) ...
+ (h/2)*(f(t(i),Vc_mod_euler(i)) ...
+ f(t(i+1),Vc_predicted));
if abs(Vc_corrected-Vc_predicted)/Vc_predicted...
<tolerance
break;
else
Vc_predicted = Vc_corrected;
end
end
% if we reached the iteration limit of 50,then print
% warning:
if j==50
fprintf('Warning at step ');
fprintf('%d: reached iteration limit\n',i);
```

```

end
Vc_mod_euler(i+1)=Vc_corrected;
end
% Plot results
subplot(1,2,1),plot(t,Vc_euler,'x',t,Vc_exact),
legend('Euler','Exact','Location','NorthWest');
title('Euler solution compared to exact');
xlabel('seconds'); ylabel('Vc (volts)');
text(0.006,0.005,sprintf('%d steps',steps));
subplot(1,2,2),plot(t,Vc_mod_euler,'x',t,Vc_exact),
legend('Modified Euler','Exact','Location','NorthWest');
title('Modified Euler solution compared to exact');
xlabel('seconds'),ylabel('Vc (volts)');
text(0.006,0.005,sprintf('%d steps',steps));
%-----

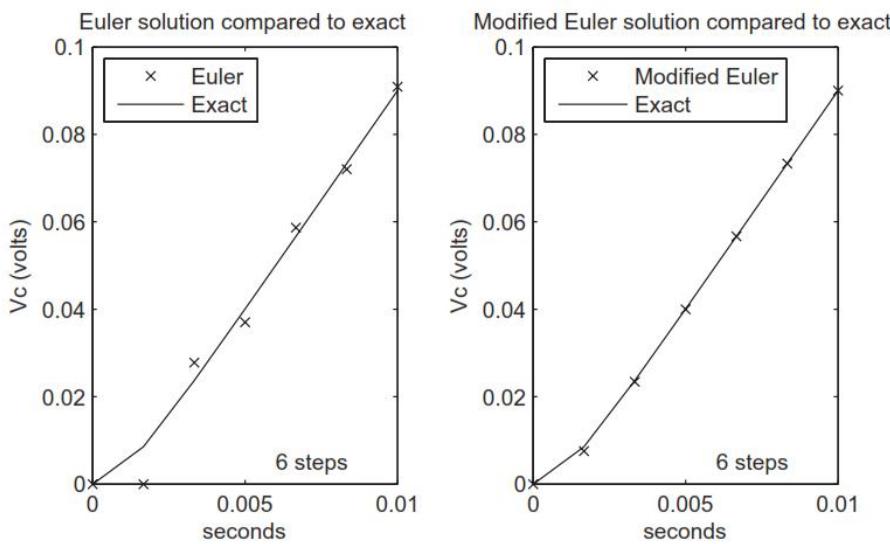
```



شکل ۵.۶: نمودار v_C در مقابل t برای مدار RC برای $20 \leq t \leq 10\tau$: (الف) جواب دقیق و روش اوولر (ب) جواب دقیق و روش اوولر اصلاح شده.

جواب حاصل برای مثال ۱.۶ در نمودار شکل ۵.۶ نشان داده شده است. توجه کنید که بازه $N = 20$ قسمت در حوزه زمان، جواب هردو روش اوولر و روش اصلاح شده تقریباً غیر قابل تشخیص از جواب دقیق است.

شکل ۵.۶ همان مثال است با این تفاوت که این دفعه ما N را به شش تقلیل داده‌ایم. (در برنامه steps=20 را با steps=6 عوض کرده‌ایم). توجه کنید که در این حالت، جواب اوولر از جواب دقیق فاصله گرفته و همچنین حول جواب دقیق نوسان می‌کند. عبارت دیگر، جواب برای روش اوولر اصلاح شده حتی با کوچک بودن N هم دقیق و پایدار است. از نظر ایده‌آل، دوست داریم N را تا حد ممکن (برای کم شدن زمان اجرای برنامه) تقلیل و کم کنیم، اما در همین هنگام نیز دوست داریم از ناپیداری عددی جلوگیری و دوری نمائیم. بنابراین، روش اوولر اصلاح شده با وجود عملیات محاسباتی اضافی (بعثت آلگوریتم پیشگوئی و تصحیح) هنوز هم برای قدمهای بزرگ پایدار است.



شکل ۶.۶: نمودار v_C در مقابل t برای مدار RC برای شش قدم در بازه $t \leq 10\tau$ (الف) جواب دقیق و روش اولر به نوسانات جواب اولر توجه کنید. (ب) جواب دقیق و روش اولر اصلاح شده. توجه کنید که روش اولر اصلاح شده حتی برای اندازه قدمهای بزرگ هم پایدار است.

۵.۶ خطای عددی آلگوریتم اولر

حل معادله دیفرانسیل با استفاده از آلگوریتم اولر و اولر اصلاح شده طبق تعریف یک راه حل تقریبی برای جواب دقیق است زیرا مشتق این آلگوریتم‌ها بر اساس سری تیلور مرخم^{۱۵} است. اگر بخواهیم دقت مخصوصی عددی داشته باشیم، می‌توانیم آنرا در معادله دیفرانسیلی که دارای جواب دقیق است امتحان کنیم. متاسفانه، در اکثر مواقع، نمیدانیم یا نمیتوانیم جواب دقیق معادله دیفرانسیل را داشته باشیم؛ بدین دلیل است که از معمولاً به روش عددی روی می‌آوریم. البته، در مثال ۱.۶، ما جواب دقیق را پیدا کردیم؛ بنابراین، می‌توانیم درصد خطای جهانی^{۱۶} را برای آلگوریتم عددی در قدم^۷ بصورت زیر بدست آوریم:

$$E_{i,global} = \left| \frac{y_{i,exact} - y_{i,numerical}}{y_{i,exact}} \right| \quad (18.6)$$

راه دیگر اندازه‌گیری خطای مرحوم محلی^{۱۷} (LTE) بنحوی که می‌توانیم جملات مرتبه بالاتر در سری تیلور (لاقل یک جمله دیگر) را محاسبه کرده و آنرا بکار برد و تضمیم بگیریم که قدمهای بیشتری داشته باشیم یا خیر. برای مثال، در معادله^(۳.۶)، ما جمله مرتبه دوم و جملات مرتبه بالاتر در سری تیلور را در روش اولر درنظر نگرفتیم؛ بنابراین LTE برابر است با

$$LTE_{Euler} = \frac{y_i}{2} h^2 +$$

اگر جملات مرتبه بالاتر را چشم‌پوشی کنیم، در این صورت خطای مرحوم از مرتبه h^2 بوده، که معمولاً بصورت $O(h^2)$ نوشته می‌شود. از نظر کمی، این بدین معنی است که اگر اندازه قدم را با ضریب دو تقلیل دهیم، در این صورت، LTE تقریباً با ضریب چهار کاهش یافته، و در نتیجه جواب دقیق‌تر حاصل خواهد شد.

^{۱۵}truncated Taylor series

^{۱۶}global percentage error

^{۱۷}local truncation error (LTE)

در عمل، روش ساده برای انتخاب اندازه قدم، با کسری از سریعترین پارامتر معلوم سیستم آغاز می‌کنیم. برای مثال، در مدار RC، ما با اندازه قدم $h = \frac{RC}{100}$ ثابت زمانی، یعنی $h = \frac{RC}{100}$. سپس، اندازه قدم را دوباره اگر دو جواب یکسان بودند، در این صورت اندازه قدم بقدر کافی کوچک است، و اگر برنامه گند اجرا می‌شود، لذا احتمالاً باید اندازه قدم را دو برابر کرده و برنامه سریعتر اجرا شده و زمان اجرا را بدون از دست دادن دقت مقایسه نمائید.

۶.۶ روش رانج کوتا مرتبه چهارم

روش رانج کوتا مرتبه چهارم از میانگین وزن دار^{۱۸} مشتق برای تخمین بین فواصل مورد نظر استفاده کرده تا مقدار y_{i+1} را محاسبه نماید. ابتدا ما با معادله دیفرانسیل مرتبه اول رابطه (۱.۶) با معلوم بودن شرایط اولیه شروع می‌کنیم. در روش اولر اصلاح شده، ما از $y_{i+1} = y_i + \frac{h}{2}(y_i + y_{i+1})$ استفاده کردیم. در روش رانج کوتا، بجای آن از

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (19.6)$$

استفاده می‌کنیم. در اینجا

$$\begin{aligned} k_1 &= f(x_i, y_i) && (\text{مقدار } y \text{ در } x_i) \\ k_2 &= f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) && (\text{تخمین } y \text{ در } x_i + \frac{h}{2}) \\ k_3 &= f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) && (\text{دوم تخمین } y \text{ در } x_i + \frac{h}{2}) \\ k_4 &= f(x_i + h, y_i + hk_3) && (\text{تخمین } y \text{ در } x_{i+1}) \end{aligned}$$

آلگوریتم رانج کوتا یک آلگوریتم صریح بوده و بنابراین محاسبه با متلب ساده است.

مثال ۲.۶:

مدار RC مثال ۱.۶ را با استفاده از روش رانج کوتا برای قدمهای $N = 20$ و $N = 6$ حل کنید. ضمناً بد ترین خطای جهانی برای این حالت را پیدا کنید. برنامه متلب بقرار زیر است:

```
% Example_6_2.m
% Calculate ramp response of RC circuit via two methods:
% 1. exact solution
% 2. Runge-Kutta method
clc; clear;
% circuit parameters
R=1e3; % ohms
C=1e-6; % farads
tau = R*C; % seconds
Vo = 10; % volts
f = @(tm,Vc) 1/tau*(Vo*tm-Vc); % Equation 6.16
% parameters for Euler algorithm
interval = 10*tau;
steps = [20 6];
for j=1:length(steps)
    h = interval/steps(j);
    % create set of time points to calculate
```

^{۱۸}weighted average

```

t = 0:h:interval;
Vc_exact = zeros(length(t),1);
Vc_RK = zeros(length(t),1);
% initial conditions
Vc_exact(1)=0;
Vc_RK(1)=0;
for i=1:length(t)-1
    % Calculate exact solution
    Vc_exact(i+1) = Vo ...
        * (t(i+1)-tau+tau*exp(-t(i+1)/tau));
    % Calculate Runge-Kutta solution
    k1=f(t(i),Vc_RK(i));
    k2=f(t(i)+h/2,Vc_RK(i)+h/2*k1);
    k3=f(t(i)+h/2,Vc_RK(i)+h/2*k2);
    k4=f(t(i+1),Vc_RK(i)+h*k3);
    Vc_RK(i+1)=Vc_RK(i)+h/6*(k1+2*k2+2*k3+k4);
end
% plot results
subplot(1,length(steps),j)
plot(t,Vc_RK,'x',t,Vc_exact),
legend('Runge-Kutta','Exact','Location','NorthWest');
title('Ramp Response for RC circuit');
xlabel('seconds'),ylabel('Vc (volts)');
text(0.005,0.005,sprintf('%d steps',steps(j)));
% calculate global error
global_error_RK = abs((Vc_exact -Vc_RK) ./ Vc_exact);
fprintf('Worst case global percent error for ');
fprintf('Runge-Kutta for %2d steps: %4.2f %%\n',...
steps(j),100*max(global_error_RK));
end
%-----

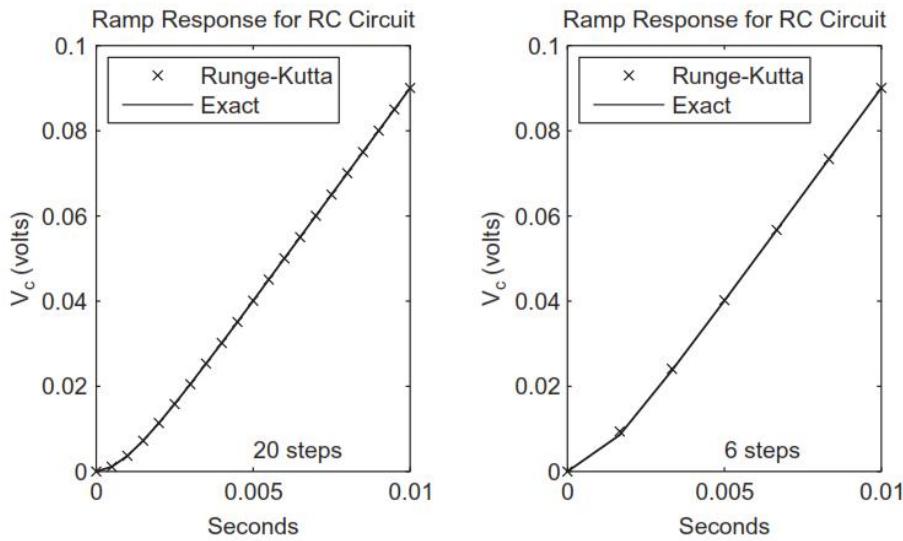
```

نتایج برنامه

بدترین در صد خطای جهانی برای رانج کوتا برای قدمهای 20 برابر 0.23% و برای قدمهای 6 برابر 9.73% است.

Worst case global percent error for Runge-Kutta for 20 steps: 0.23%
 Worst case global percent error for Runge-Kutta for 6 steps: 9.73%

توجه کنید که هر قدر قدمها کمتر شود، خطای جهانی مربوطه بیشتر می‌شود. این نتایج در شکل (۷.۶) نشان داده شده است.



شکل ۷.۶: نمودار v_C در مقابل t برای مدار RC برای روش رانج کوتا: (الف) بیست قدم (ب) شش قدم.

۷.۶ دستگاه دو معادله دیفرانسیل مرتبه اول

دو معادله دیفرانسیل مرتبه اول زیر را در نظر گیرید

$$\begin{aligned} \frac{du}{dt} &= f(t, u, v); & u(0) = u_0 \\ \frac{dv}{dt} &= g(t, u, v); & v(0) = v_0 \end{aligned} \quad (۲۰.۶)$$

برای حل دو معادله دیفرانسیل مرتبه دوم با روش رانج کوتا، فرض کنید

$$\begin{aligned} u_{i+1} &= u_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ v_{i+1} &= v_i + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4) \end{aligned} \quad (۲۱.۶)$$

که در آن

$$\begin{aligned} k_1 &= f(t_i, x_i, y_i) & \text{(تخمین } u \text{ در } (t_i)) \\ l_1 &= g(t_i, x_i, y_i) & \text{(تخمین } v \text{ در } (t_i)) \end{aligned}$$

$$\begin{aligned} k_2 &= f\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_1, v_i + \frac{h}{2}l_1\right) & \text{(تخمین } u \text{ در } (t_i + \frac{h}{2})) \\ l_2 &= g\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_1, v_i + \frac{h}{2}l_1\right) & \text{(تخمین } v \text{ در } (t_i + \frac{h}{2})) \end{aligned}$$

$$\begin{aligned} k_3 &= f\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_2, v_i + \frac{h}{2}l_2\right) & \text{(دوم تخمین } u \text{ در } (t_i + \frac{h}{2})) \\ l_3 &= g\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_2, v_i + \frac{h}{2}l_2\right) & \text{(دوم تخمین } v \text{ در } (t_i + \frac{h}{2})) \end{aligned}$$

$$\begin{aligned} k_4 &= f(t_i + h, u_i + hk_3, v_i + hl_3) & \text{(تخمین } u \text{ در } (t_i + h)) \\ l_4 &= g(t_i + h, u_i + hk_3, v_i + hl_3) & \text{(تخمین } v \text{ در } (t_i + h)) \end{aligned} \quad (۲۲.۶)$$

و $h = \Delta t = t_{i+1} - t_i$ است.

مثال ۳.۶:

دستگاه دو معادله دیفرانسیل زیر را با روش رانج کوتا در بازه $0 \leq t \leq 10$ حل کنید:

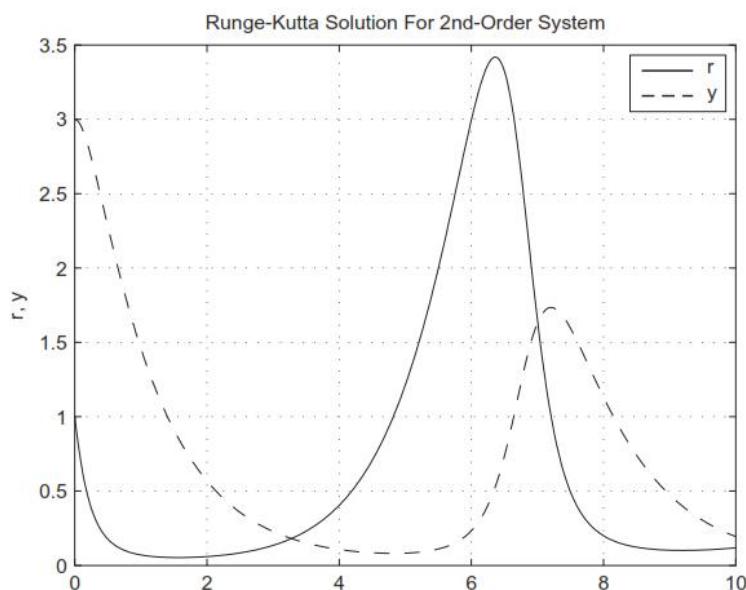
$$\begin{aligned}\frac{dr}{dt} &= 2re^{-0.1t} - 2ry = f(t, r, y) \\ \frac{dy}{dt} &= -y + ry = g(r, y)\end{aligned}\quad (۲۳.۶)$$

شرط اولیه: $r(0) = 1.0, y(0) = 3.0$ است.

```
% Example_6_3.m
% This program solves a system of 2 first order ordinary
% differential equations by the Runge-Kutta Method.
% First, define two functions to compute dr/dt and dy/dt:
rprime_func = @(t,r,y) 2.0*r*exp(-0.1*t) -2.0*r*y;
yprime_func = @(r,y) -y + r*y;
% Open a file to write results:
fo=fopen('Example_6_3_output.txt','w');
fprintf(fo,'t r y \n');
fprintf(fo,'-----\n');
% Define time interval and step size
tmax=10; steps=1000; h=tmax/steps;
% Initial conditions:
r(1)=1.0; y(1)=3.0; t(1)=0.0;
% Estimate of derivatives and marching in time.
for i=1:steps
    t(i+1)=i*h;
    K(1)=rprime_func(t(i),r(i),y(i));
    L(1)=yprime_func(r(i),y(i));
    K(2)=rprime_func(t(i)+h/2,r(i)+h/2*K(1),...
        y(i)+h/2*L(1));
    L(2)=yprime_func(r(i)+h/2*K(1),y(i)+h/2*L(1));
    K(3)=rprime_func(t(i)+h/2,r(i)+h/2*K(2),...
        y(i)+h/2*L(2));
    L(3)=yprime_func(r(i)+h/2*K(2),y(i)+h/2*L(2));
    K(4)=rprime_func(t(i)+h,r(i)+h*K(3),y(i)+h*L(3));
    L(4)=yprime_func(r(i)+h*K(3),y(i)+h*L(3));
    r(i+1)=r(i)+h/6*(K(1)+2*K(2)+2*K(3)+K(4));
    y(i+1)=y(i)+h/6*(L(1)+2*L(2)+2*L(3)+L(4));
    if mod(i-1,10)==0
        fprintf(fo,'%.2f %10.4f %10.4f\n',t(i),r(i),y(i));
    end
end
fclose(fo);
plot(t,r,t,y,'--'), xlabel('t'), ylabel('r,y');
grid, legend('r', 'y');
title('Runge-Kutta Solution For 2nd-Order System');
-----
```

نتایج برنامه

نتایج نمودار متلب در شکل ۸.۶ نشان داده شده است.



شکل ۸.۶: جواب دو معادله دیفرانسیل مرتبه اول توسط روش رانج کوتا.

۸.۶ معادله مرتبه دوم

برای معادله دیفرانسیل معمولی مرتبه دوم، روش حل تقلیل معادله بیک سیستم دو معادله دیفرانسیل مرتبه اول است. فرض کنید معادله دیفرانسیل مرتبه دوم زیر با شرایط اولیه داده شده است.

$$\begin{aligned} u'' &= \frac{d^2u}{dt^2} = f(t, u, u') \\ u(0) &= u_0 \quad \text{و} \quad u'(0) = u_0 \end{aligned} \quad (24.6)$$

فرض کنید $v = u'$ باشد، در این صورت $u'' = \frac{dv}{dt} = v = f(t, u, v)$ است. همچنین، v میدهد

$$\begin{aligned} u' &= v \\ v &= f(t, u, v) \end{aligned} \quad (25.6)$$

بنابراین، ما معادله (۲۴.۶) را به دو معادله دیفرانسیل مرتبه اول بفرم معادله (۲۰.۶) تبدیل کردیم؛ لذا، همان روش راه حل را می‌توان اعمال کرد، یعنی

$$\begin{aligned} k_1 &= f(t_i, u_i, v_i) & (v' \text{ در } t_i) \\ l_1 &= u'(t_1) = v_i & (u' \text{ در } t_i) \end{aligned}$$

$$k_2 = f \left(t_i + \frac{h}{2}, u_i + \frac{h}{2} l_1, v_i + \frac{h}{2} k_1 \right) \quad (\text{تخمین اول } v \text{ در } t_i + \frac{h}{2})$$

$$l_2 = u \left(t_i + \frac{h}{2} \right) = v_i + \frac{h}{2} k_1 \quad (\text{تخمین اول } u' \text{ در } t_i + \frac{h}{2})$$

$$k_3 = f \left(t_i + \frac{h}{2}, u_i + \frac{h}{2} l_2, v_i + \frac{h}{2} k_2 \right) \quad (\text{دوم تخمین } u \text{ در } t_i + \frac{h}{2})$$

$$l_3 = u \left(t_i + \frac{h}{2} \right) = v_i + \frac{h}{2} k_2 \quad (\text{دوم تخمین } u' \text{ در } t_i + \frac{h}{2})$$

$$k_4 = f(t_i + h, u_i + h l_3, v_i + h k_3) \quad (\text{تخمین } v \text{ در } t_i + h)$$

$$l_4 = u(t_i + h) = v_i + h k_3 \quad (\text{تخمین } u' \text{ در } t_i + h)$$

مقادیر u و v در قدم زمانی بعدی چنین است

$$u_{i+1} = u_i + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4)$$

$$v_{i+1} = v_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

مثال ۴.۶

این روش را با اعمال بر مدار سری RLC پیوست الف تشریح و روشن می‌کنیم. معادله (الف.۶) ولتاژ خازن v_C (بدون ولتاژ تغذیه) بصورت زیر میدهد

$$\frac{d^2v_C}{dt^2} + \frac{R}{L} \frac{dv_C}{dt} + \frac{1}{LC} v_C = 0$$

می‌خواهیم v_C را بازه Ω حل کنیم. $\frac{dv_C}{dt}(0) = \frac{6}{\sqrt{LC}}$

ابتدا این معادله دیفرانسیل مرتبه دوم را بدو معادله مرتبه اول تقلیل میدهیم.

فرض کنید $v_2 = \frac{dv_C}{dt}$ باشد. در این صورت، دو معادله کوپل شده برابرند با

$$\frac{dv_C}{dt} = v_2$$

$$\frac{dv_2}{dt} = \frac{d^2v_C}{dt^2} = -\frac{1}{LC}v_C - \frac{R}{L}v_2$$

فرض کنید سیستم زیرمیرائی است، جواب دقیق بصورت زیر خواهد بود

$$v_c = \exp\left(-\frac{R}{2L}t\right) \left[\alpha \cos\left(\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} t\right) + \beta \sin\left(\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} t\right) \right]$$

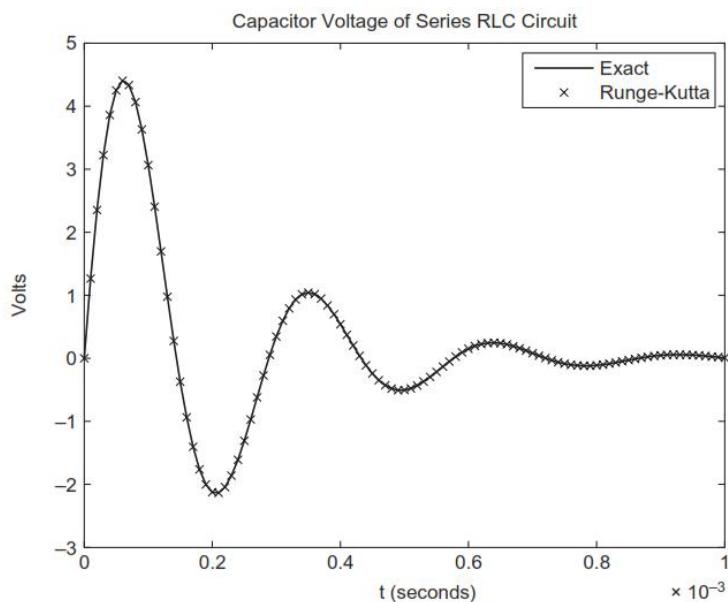
برای شرایط اولیه داده شده $\alpha = 0$ و $\beta = \frac{\frac{dv_2}{dt}(0)}{\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2}}$ بدست می‌آید.

برنامه در زیر آورده شده است:

```
% Example_6_4.m
% This program solves a single second order ordinary
% differential equation by the Runge-Kutta Method. The
% differential equation describes the capacitor voltage
% in a series RLC circuit
clear; clc;
```

```
% Component values:
R=10; L=1.0e-3; C=2.0e-6;
% Initial conditions:
vc(1)=0; v2(1)=6* sqrt(1/(L*C));
% Define time interval and step size. Since the natural
% frequency sqrt(1/LC) is 3559 Hz,choose a time interval
% to capture several oscillations. 3559 Hz has period of
% 281 usec,so choose 1000 usec for interval.
tmax=1000e-6; steps=100; h=tmax/steps;
t=0:h:tmax;
% Define v2'= f(t,x,y):
v2_prime = @(t,x,y) -(1/(L*C))*x -(R/L)*y;
% Do Runge-Kutta algorithm:
for i=1:steps
K1=v2_prime(t(i),vc(i),v2(i));
L1=v2(i);
K2=v2_prime(t(i),vc(i)+h/2*L1,v2(i)+h/2*K1);
L2=v2(i)+h/2*K1;
K3=v2_prime(t(i),vc(i)+h/2*L2,v2(i)+h/2*K2);
L3=v2(i)+h/2*K2;
K4=v2_prime(t(i),vc(i)+h*L3,v2(i)+h*K3);
L4=v2(i)+h*K3;
vc(i+1)=vc(i)+h/6*(L1+2*L2+2*L3+L4);
v2(i+1)=v2(i)+h/6*(K1+2*K2+2*K3+K4);
end
% Calculate exact solution
beta = v2(1) / sqrt( 1/(L*C) -(R/(2*L))^2 );
vc_exact = exp(-R/(2*L) * t)* beta ...
.* sin(sqrt( 1/(L*C) -(R/(2*L))^2)*t );
% Print results to screen.
fprintf('Runge-Kutta solution for a single ');
fprintf('second order ODE:\n\n');
fprintf('t R-K Sol''n Exact Sol''n\n');
fprintf('-----\n');
for i=1:10:steps+1 % just print every tenth step
fprintf('%10.3e %8.4f %8.4f \n',...
t(i),vc(i),vc_exact(i));
end
% Plot results:
plot(t,vc_exact,t,vc,'x');
xlabel('t (seconds)'),ylabel('volts');
title('Capacitor voltage of series RLC circuit');
legend('Exact','Runge-Kutta');
%-----
```

مقایسه جواب دقیق و جواب رانج کوتا در شکل (۹.۶) نشان داده شده است.



شکل ۹.۶: جواب معادله دیفرانسیل مرتبه دوم با روش رانج کوتا.

۹.۶ تابع معادله دیفرانسیل معمولی متلب

متلب چندین تابع معادله دیفرانسیل معمولی دارد که یک سیستم معادله دیفرانسیل معمولی مرتبه اول را حل می‌کند، آن شامل ode23 و ode45 است. در این فصل، ما ode45 را را نشانده‌داره که آن بر اساس روش رانج کوتای مرتبه چهارم و پنجم است. توضیح تابع ode45 بقرار زیر است (شما می‌توانید با نوشتتن help در پنجره فرمان متلب توضیحات را بدست آورید):

```
ODE45 Solve non-stiff differential equations,medium order method.
[TOUT,YOUT] = ODE45(ODEFUN,TSPAN,Y0) with TSPAN = [T0
TFINAL] integrates the system of differential equations y'=
f(t,y) from time T0 to TFINAL with initial conditions Y0.
ODEFUN is a function handle. For a scalar T and a vector Y,
ODEFUN(T,Y) must return a column vector corresponding to
f(t,y). Each row in the solution array YOUT corresponds to
a time returned in the column vector TOUT. To obtain
solutions at specific times T0,T1,...,TFINAL (all
increasing or all decreasing),use TSPAN = [T0 T1...
TFINAL].
```

بنابراین، آرگومانهای را گرفته و آنرا به (ODEFUN) که تشریح کننده معادله دیفرانسیل است، و بردار (TSPAN) که تشریح کننده بازه زمانی است، و بردار Y0 که تشریح کننده شرایط اولیه است می‌دهد. تابع ODEFUN باید دو آرگومان داشته باشد: زمان t و یک بردار شامل مقادیر y_1, y_2, \dots, y_n که معادله دیفرانسیل برساند. دستگاه n معادله دیفرانسیل باید بصورت استاندارد باشد، یعنی، n معادله بصورت $y_n = f(t, y_1, y_2, \dots, y_n)$ باشد. ODEFUN باید یک بردار از مشتقهای y_1, y_2, \dots, y_n باشد. بازه زمانی TSPAN معمولاً یک بردار دو عنصره شامل زمان ابتدا و انتهای است؛ ode45 بطور خودکار قدمهای

زمانی مناسب را انتخاب (و قدمهای زمانی ممکن است در بازه زمانی تغییر کند) می‌کند. بردار بر میگرداند: یک از نقاط زمانی TOUT و جواب در هر نقطه YOUT. اگر می‌خواهید ode45 را مجبور کنید دستگاه را در نقاط زمانی مشخصی حل کند، باید بطورصریح نقاط زمانی را در TSPAN (بجای دادن نقطه اول و آخر) مشخص کنید.

مثال ۵.۶

دستگاه سه معادله دیفرانسیلی زیر را با استفاده از ode45 مطلب حل کنید:

$$\begin{aligned}y_1 &= y_2 y_3 t \\y_2 &= -y_1 y_3 \\y_3 &= -0.5 y_1 y_2\end{aligned}$$

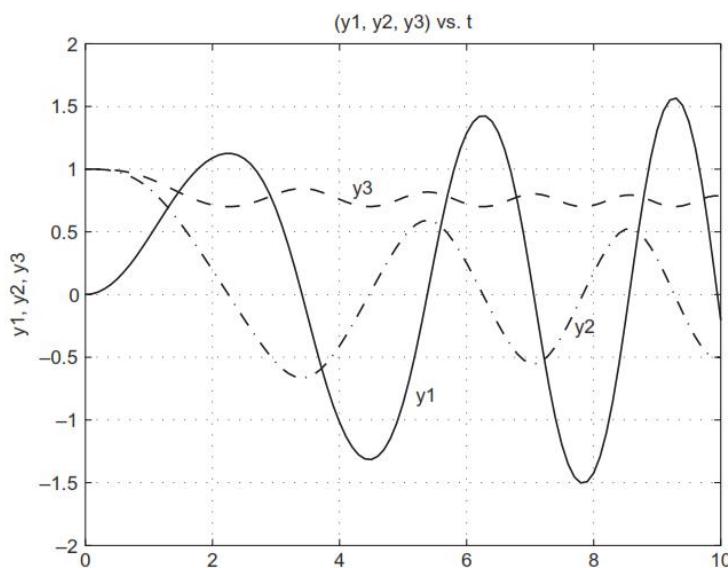
شرط اولیه چنین است: $y_3(0) = 1.0$ ، $y_2(0) = 1.0$ و $y_1(0) = 0$

```
% Example_6_5.m
% This program solves a system of 3 ordinary differential
% equations by using MATLAB's ode45 function.
% y1'=y2*y3*t,y2'=-y1*y3,y3'=-0.51*y1*y2
% y1(0)=0,y2(0)=1.0,y3(0)=1.0
clear; clc;
initial=[0.0 1.0 1.0];
tspan=0.0:0.1:10.0;
[t,Y]=ode45(@dydt3,tspan,initial);
y1=Y(:,1);
y2=Y(:,2);
y3=Y(:,3);
fid=fopen('output.txt','w');
fprintf(fid,'t y1 y2 y3 \n');
fprintf(fid,'-----\n');
for i=1:2:101
    fprintf(fid,'%6.2f %8.4f %8.4f %8.4f \n',...
    t(i),y1(i),y2(i),y3(i));
end
fclose(fid);
plot(t,y1,t,y2,'-.',t,y3,'--');
xlabel('t'), ylabel('y1,y2,y3');
title('(y1,y2,y3) vs. t'), grid;
text(5.2,-0.8,'y1'), text(7.7,-0.25,'y2');
text(4.2,0.85,'y3');
%-----
% dydt3.m
% This function works with Example_6_5.m
% y1'=y2*y3*t,y2'=-y1*y3,y3'=-0.51*y1*y2
% y1=Y(1),y2=Y(2),y3=Y(3).
function Yprime=dydt3(t,Y)
Yprime=zeros(3,1);
Yprime(1)=Y(2)*Y(3)*t;
```

```
Yprime(2)=-Y(1)*Y(3);  
Yprime(3)=-0.51*Y(1)*Y(2);
```

نتایج برنامه

نتایج محاسبه شده برای y_1 , y_2 و y_3 در شکل ۱۰.۶ نشان داده شده است.



شکل ۱۰.۶: جواب سه معادله دیفرانسیل با استفاده از تابع ode45 متلب.

مثال ۶.۶:

مدار RLC بخش ۱۷.۲ را با استفاده از ode45 متلب حل کنید. فرض کنید $L = 1\mu H$, $R = 50\Omega$, $C = 10nF$, $v(0) = 3.3 V$, $i_L(0) = 0 A$ و $0 \leq t \leq 4\mu s$ است. منحنی نتایج را برای بازه زمانی ۰ رسم کنید.

```
% Example_6_6.m  
% Solve a system of two first-order ordinary differential  
% equations (Equations 2.6 and 2.7).  
% Assume: y(1)=v and y(2)=iL  
% Assume: R=50,L=1e-6,C=10e-9,v(t=0)=3.3,i(t=0)=0.  
% Equation 2.6: yprime(2) = (1/L)*y(1)  
% Equation 2.7: yprime(1) = -1/(R*C)*y(1) -(1/C)*y(2)  
clear; clc;  
initial=[3.3 0];  
tspan=0:.01e-6:4e-6;  
[ t,y ] = ode45('dydt_rlc',tspan,initial);  
v = y(:,1);  
iL = y(:,2);
```

```

fprintf('t v iL \n');
fprintf('-----\n');
% print every tenth value of v and iL
for i=1:10:length(t)
fprintf('%1.10f %10.4f %10.4f \n',...
t(i),v(i),iL(i));
end
% Plot v and iL together. Use plotyy() to create % second 'y'axis on the
right side. plotyy() returns
% graphics handles to the axes and lines so that their
% style can be adjusted.
[ axis,line1,line2 ] = plotyy(t,v,t,iL);
xlabel('t'),ylabel('v (volts)');
title('v and iL vs. t'),grid;
% set label for right 'y'axis
set(get(axis(2),'Ylabel'),'String','i_L (amps)');
% plot iL with a dashed line
set(line2,'LineStyle','--');
legend('v','i_L');
%-----
% dydt_rlc.m
% This function works with Example_6_6.m
% Assume: v=y(1) and iL=y(2)
function yprime = dydt_rlc(t,y)
yprime=zeros(2,1);
R=50; L=1e-6; C=10e-9;
yprime(1) = -1/(R*C)*y(1) -1/C*y(2);
yprime(2) = (1/L)*y(1);

```

نتایج برنامه

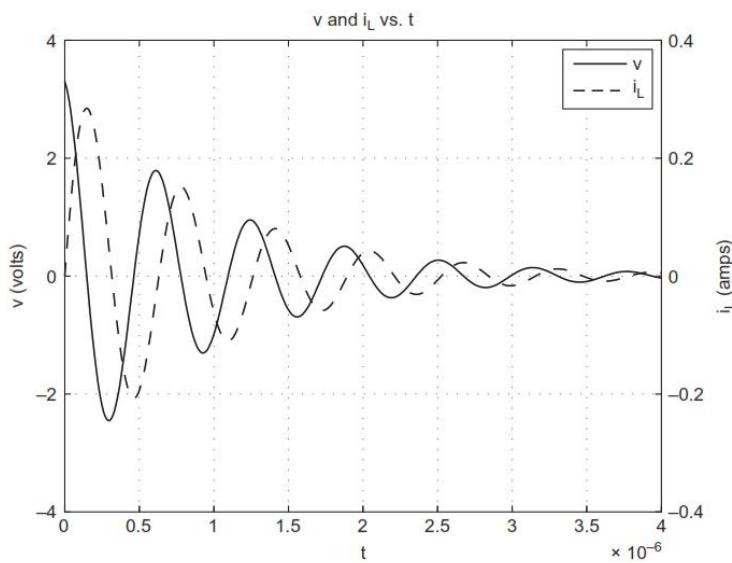
نتایج در شکل (۱۱.۶) رسم شده است. توجه کنید که طبیعت تکمیلی ($v(t)$ و $i_L(t)$) وقتی هر کدام صفر هستند، دیگری همواره نزدیک بیشینه است. این رفتار نشان میدهد که چگونه انرژی در مدار RLC نوسان رفت و برگشتی در خازن و سیمپیج دارد.

۱۰.۶ مسائل مقدار مرزی

وقتی یک معادله دیفرانسیل معمولی ODE شامل شرایط مرزی بجای شرایط اولیه دارد، در این صورت ما برای حل مسئله رویکرد دیگری بکار می‌بریم. در مسئله مقدار مرزی، اصولاً لازم است جواب مسئله را به شرایط مرزی "تطبیق"^{۱۹} دهیم که این مخالف انتگرال‌گیری ساده از شرایط اولیه است. بعنوان مثال، این نوع مسئله تعیین میدان الکتریکی بین دو صفحه یک خازن با چگالی بار معلوم در روی صفحات و ولتاژ ثابت روی آنها است. ولتاژ ثابت شرایط مرزی، و جواب را میتوان از حل عددی معادله مرتبه دوم، غیر همگن معادله دیفرانسیل معمولی با استفاده از روش تفاضلات محدود^{۲۰} بدست آورد. با شرایط

^{۱۹}fit

^{۲۰}finite difference method

شکل ۱۱.۶: جواب مدار RLC با استفاده از تابع `ode45` متلب.

مرزی مخصوص، روش عددی معادله دیفرانسیل را به حل یک دستگاه معادلات جبری خطی که به ماتریس سه-قطري^{۲۱} معروف است تبدیل می‌کند. جواب معادلات جبری خطی که شامل محاسبات ساده‌تری نسب به روش حذف گوسی است در فصل سوم تشریح شد. این برای حل دستگاه معادلات بزرگ خیلی مهم است. حل دستگاه سه-قطري بعداً بحث می‌شود.

۱۱.۶ حل دستگاه سه-قطري معادلات خطی

یک دستگاه معادلات سه-قطري بصورت زير است:

$$\begin{bmatrix} 1 & a_1 & 0 & 0 & 0 \\ -b_2 & 1 & -a_2 & 0 & 0 \\ 0 & -b_3 & 1 & -a_3 & 0 \\ 0 & 0 & -b_4 & 1 & -a_4 \\ 0 & 0 & 0 & -b_5 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} \quad (26.6)$$

که در آن a_i و b_i و c_i جملات ثابت و x_i بدان نامیده می‌شود که عناصر ضرایب ماتریس، در قسمت سمت چپ معادله (۲۶.۶) در سه قطر قرار دارند و قطر اصلی آن بدون صفر است. سایر عناصر غیر این سه قطر همگی صفر هستند. با ضرب ماتریس‌ها در این معادله (۲۶.۶)، مجموعه معادلات برای این دستگاه مرتبه پنجم خواهد شد:

$$x_1 - a_1 x_2 = c_1 \quad (27.6)$$

$$-b_2 x_1 + x_2 - a_2 x_3 = c_2 \quad (28.6)$$

$$-b_3 x_2 + x_3 - a_3 x_4 = c_3 \quad (29.6)$$

^{۲۱}tri-diagonal matrix

^{۲۲}tri-diagonal

$$-b_4x_3 + x_4 - a_4x_5 = c_4 \quad (۳۰.۶)$$

$$-b_5x_4 + x_5 = c_5 \quad (۳۱.۶)$$

مفهوم کلی برای حل یک دستگاه سه-قطري بقرار زیر است:
 می‌توانیم معادله (۲۷.۶) را برای x_1 حل کرده و نتیجه را در معادله (۲۸.۶) قرار دهیم، حال یک معادله‌ای شامل x_2 و x_3 خواهیم داشت که نام آنرا معادله (۲۸.۶) می‌گذاریم.
 می‌توانیم معادله (۲۸.۶) را برای x_2 بر حسب x_3 حل کرده و آنرا در معادله (۲۹.۶) جایگزین کنیم. این یک معادله که فقط شامل x_3 و x_4 بما می‌دهد و نام آنرا (۲۹.۶) می‌گذاریم.
 این فرآیند را تا آخرین معادله ادامه میدهیم. وقتی x_4 را در معادله (۳۱.۶) قرار میدهیم، فقط یک معادله بر حسب x_5 خواهیم داشت و از این‌رو بما این اجازه را میدهد که از نظر جبری x_5 را بدست آوریم.
 بنابراین، با جایگذاری معکوس، می‌توانیم تمام مقادیر x_i را تعیین نمائیم.

روش متقارن برای m معادله

ترتیب مجموعه معادلات بصورت کلی زیر است

$$x_i = a_i x_{i+1} + b_i x_{i-1} + c_i \quad (۳۲.۶)$$

توجه: برای اولین و آخرین معادله (مربوط به ضرایب بالای چپ و راست در ماتریس سه-قطري) $b_1 = 0$ و $a_m = 0$ هستند.

با قرار دادن دستورالعمل خلاصه شده، می‌توانیم مجموعه معادلاتی بصورت زیر داشته باشیم

$$x_i = d_i + e_i x_{i+1} \quad (۳۳.۶)$$

که در آن d_i و e_i ضرایب معادلات پریم دار هستند. توجه کنید که $e_m = 0$ است. در این صورت،

$$\begin{aligned} x_m &= d_m \\ x_{m-1} &= d_{m-1} + e_{m-1} x_m \\ &\vdots \\ x_1 &= d_1 + e_1 x_2 \end{aligned} \quad (۳۴.۶)$$

اگر بتوان رابطه کلی برای d_i و e_i بدست آورد، آنوقت می‌توان دستگاه را برای x_i حل کرد.
 ما با بازنویسی رابطه (۳۳.۶) شروع کرده و $i = 1, 2, \dots, m$ می‌توانیم معادله را بصورت زیر می‌نویسیم

$$x_{i-1} = d_{i-1} + e_{i-1} x_i \quad (۳۵.۶)$$

با قرار دادن معادله (۳۵.۶) در (۳۲.۶) خواهیم داشت:

$$x_i = a_i x_{i+1} + b_i (d_{i-1} + e_{i-1} x_i) + c_i$$

و حل آن برای x_i میدهد

$$x_i = \frac{(c_i + b_i d_{i-1})}{(1 - b_i e_{i-1})} + \frac{a_i x_{i+1}}{(1 - b_i e_{i-1})} \quad (۳۶.۶)$$

جملات پیشرونده بین معادلات (۳۶.۶) و (۳۲.۶) خواهد بود

$$\begin{aligned} d_i &= \frac{c_i + b_i d_{i-1}}{1 - b_i e_{i-1}} \\ e_i &= \frac{a_i}{1 - b_i e_{i-1}} \end{aligned} \quad (۳۷.۶)$$

که آن برای $i = 2, 3, \dots, m$ درست است.

فصل ۶. انتگرال‌گیری عددی معادلات دیفرانسیل معمولی

۱۷۷

خیلی از اولین معادله در یک دستگاه بصورت $x_i = d_i + e_i x_{i-1}$ است؛ بنابراین، جملات پیشرونده بین معادلات (۳۷.۶) و (۳۳.۶)، خواهد بود

$$\begin{aligned} d_1 &= c_1 \\ e_1 &= a_1 \end{aligned} \quad (38.6)$$

اگر نمی‌توانیم پشت سر هم بر معادله (۳۷.۶) اعمال کرد و d_2, d_3, \dots, d_m و e_2, e_3, \dots, e_m را بنابراین، $x_m = d_m$ با جایگذاری معکوس بدست آوریم.

$$\begin{aligned} x_{m_1} &= d_{m-1} + e_{m-1} x_m \\ x_{m_1} &= d_{m-1} + e_{m-1} x_m \\ &\vdots \\ x_1 &= d_1 + e_1 x_2 \end{aligned}$$

: مثال ۷.۶

دستگاه معادلات زیر را برای x_i حل کنید:

$$x_1 + 2x_2 = 7 \quad (39.6)$$

$$4x_1 + 3x_2 + 14x_3 = -9 \quad (40.6)$$

$$11x_2 + x_3 - 6x_4 = 3 \quad (41.6)$$

$$-3x_3 + 2x_4 + 31x_5 = 1 \quad (42.6)$$

$$4x_4 + x_5 - 8x_6 = -19 \quad (43.6)$$

$$20x_5 + x_6 = 11 \quad (44.6)$$

ابتدا لازم است معادله را بصورت معادله (۳۲.۶) در آوریم، که خواهد بود

$$x_i = a_i x_{i+1} + b_i x_{i-1} + c_i$$

این میدهد

$$\begin{aligned} x_1 &= -2x_2 + 7 \\ x_2 &= -\frac{14}{3}x_3 - \frac{4}{3}x_1 - \frac{9}{3} \\ x_3 &= 6x_4 - 11x_2 + 3 \\ x_4 &= -\frac{31}{2}x_5 + \frac{3}{2}x_3 + \frac{1}{2} \\ x_5 &= 8x_6 - 4x_4 - 19 \\ x_6 &= -20x_5 + 11 \end{aligned}$$

مشاهده می‌شود که

$$\begin{aligned} a &= -2 & -14/3 & 6 & -31/2 & 8 & 0 \\ b &= 0 & -4/3 & -11 & 3/2 & -4 & -20 \\ c &= 7 & -3 & 3 & 1/2 & -19 & 11 \end{aligned}$$

برنامه مطلب برای حل معادله (۳۹.۶) برای تمام x_i بقرار زیر است:

```
% Example_6_7.m
% Example of solving a tri-diagonal system of equations
a = [-2 -14/3 6 -31/2 8 0];
b = [0 -4/3 -11 3/2 -4 -20];
c = [7 -3 3 1/2 -19 11];
```

```

m = length(a);
% Compute d and e coefficients:
d(1)=c(1);
e(1)=a(1);
for i=2:m
    d(i) = (c(i) + b(i)*d(i-1)) / (1 -b(i)*e(i-1));
    e(i) = a(i) / (1 -b(i)*e(i-1));
end
% Compute x:
x(m)=d(m);
for i=(m-1):-1:1
    x(i) = d(i) + e(i)*x(i+1);
end
% Display the solution:
x

```

نتایج برنامه

```

x =
37.3362 -15.1681 -8.0600 -29.6515 1.1653 -12.3051

```

۱۲.۶ روش تفاضلات محدود

برای حل عددی مسائل مقدار مرزی شامل معادله دیفرانسیل خطی معمولی، لازم است فرمول تفاضل را از طریق بسط سری تیلور بدست آوریم. این بما اجازه می‌دهد تا معادلات دیفرانسیل را به یک مجموعه معادلات جری خطی تبدیل کنیم. بطوری که در بخش ۳.۶ دیدیم، میتوانیم $y = g(x)$ حول نقطه x_i توسط سری تیلور بسط دهیم؛ یعنی،

$$y(x) = y(x_i) + y'(x_i)(x_i - x) + \frac{y''(x_i)}{2!}(x_i - x)^2 + \frac{y'''(x_i)}{3!}(x_i - x)^3 + \dots$$

اندازه قدم را $h = x_i - x$ تعریف می‌کنیم، که میدهد

$$y(x_i + h) = y(x_i) + y'(x_i)h + \frac{y''(x_i)}{2!}h^2 + \frac{y'''(x_i)}{3!}h^3 + \dots$$

فرض کنید $y'(x_i) = y'_i$ و $y(x_i) = y_i$ و $y(x_i + h) = y_{i+1}$ و غیره باشد. در این صورت، بسط سری تیلور را می‌توان بصورت زیر نوشت

$$y_{i+1} = y_i + y_i h + \frac{y'_i h^2}{2!} + \frac{y''_i h^3}{3!} + \dots \quad (45.6)$$

اکنون معادله (۴۵.۶) را برای نقطه $x = x_i - h$ می‌نویسیم، یعنی

$$y(x_i - h) = y_{i-1} = y_i + y_i(-h) + \frac{y'_i}{2!}(-h)^2 + \frac{y''_i}{3!}(-h)^3 + \dots$$

یا

$$y_{i-1} = y_i - y_i h + \frac{y'_i h^2}{2!} - \frac{y''_i h^3}{3!} + \dots - \dots \quad (46.6)$$

از رابطه (۴۵.۶) و (۴۶.۶) برای بدست آوردن چندین فرمول تفاضل استفاده می‌کنیم. با استفاده از اولین دو جمله رابطه (۴۵.۶) داریم:

$$y_{i+1} = y_i + y'_i h$$

جدول ۱.۶:

| | |
|---|--|
| $y'_i = \frac{y_{i+1} - y_i}{h}$ | فرمول تفاضل پیشرو مرتبه اول این برای شرط مرزی y'_i در ابتداء حوزه بکار می‌رود. |
| $y'_i = \frac{y_i - y_{i-1}}{h}$ | فرمول تفاضل پسرو مرتبه اول این برای شرط مرزی y'_i در انتهای حوزه بکار می‌رود. |
| $y'_i = \frac{y_{i+1} - y_{i-1}}{2h}$ | فرمول تفاضل مرکزی مرتبه اول این برای معادله تفاضلی مرتبه اول در نقاط داخلی بکار می‌رود. |
| $y'_i = \frac{y_{i+1} - y_{i-1} - 2y_i}{h^2}$ | فرمول تفاضل مرکزی مرتبه اول این برای معادله تفاضلی مرتبه دوم در نقاط داخلی بکار می‌رود. |

و از حل آن برای y'_i خواهیم داشت

$$y'_i = \frac{y_{i+1} - y_i}{h} \quad (47.6)$$

این فرمول تفاضل پیشرو برای y'_i با خطای مرتبه h است. بهمین نحو، از رابطه (۴۶.۶)، و استفاده از اولین دو جمله رابطه بسط تیلور خواهیم داشت

$$y'_i = \frac{y_i - y_{i-1}}{h} \quad (48.6)$$

این فرمول تفاضل پسرو برای y'_i با خطای مرتبه h است.
اکنون، بیایید رابطه (۴۶.۶) را از رابطه (۴۵.۶) کم کنیم و اولین سه جمله بسط تیلور را در نظر گیریم. این میدهد

$$y_{i+1} - y_{i-1} = 2y'_ih$$

و با حل آن برای y'_i خواهیم داشت

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h} \quad (49.6)$$

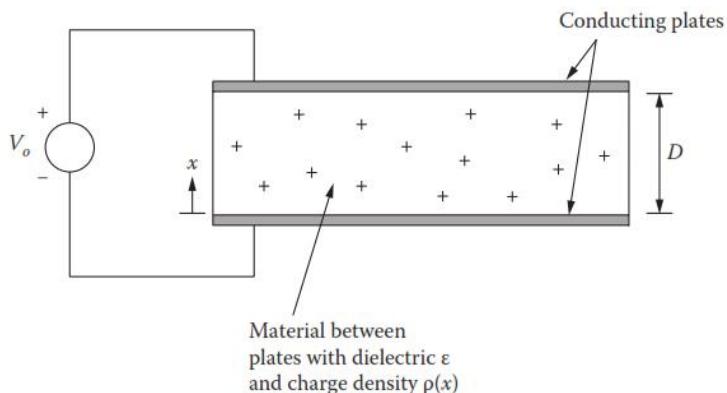
این فرمول تفاضل مرکزی برای y'_i با خطای مرتبه h^2 است. (خطا از مرتبه h^2 است، زیرا سه جمله از بسط سری تیلور برای این فرمول بکار رفته است). اگر رابطه (۴۵.۶) و (۴۶.۶) را با هم جمع کرده و اولین سه جمله را در رابطه نگه داریم، خواهیم داشت

$$y_{i+1} + y_{i-1} = 2y_i + y'_ih^2$$

و از حل آن برای y'_i خواهیم داشت

$$y'_i = \frac{y_{i+1} + y_{i-1} - 2y_i}{h^2} \quad (50.6)$$

این فرمول تفاضل مرکزی برای y'_i با خطای مرتبه h^2 است.
این چهار فرمول تفاضل در جدول (۱۲.۶) خلاصه شده است. اکنون می‌خواهیم نشان دهیم که چگونه آنها را در مسائل مقدار مرزی که منجر به یک دستگاه معادلات خطی سه قطعی می‌شود، اعمال کنیم.



شکل ۱۲.۶: خازن صفحات موازی با فاصله صفحات D متر و چگالی بار ثابت $\rho(x) C/m^3$ بین سطوح.

۱۳.۶ خازن صفحه یک بعدی

شکل (۱۲.۶) یک خازن با صفحات موازی را که ولتاژ ثابت V_0 به آن اعمال شده و چگالی بار ثابت $\rho(x) C/m^3$ در بین صفحات است را نشان می‌دهد. برای حالت‌های با تقارن، از قبیل خازن با صفحات موازی که چگالی بار فقط نسبت به x تغییر می‌کند، (مستقل از y و z است)، معادله پواسون پتانسیل الکتریکی Φ بیک معادله دیفرانسیل ختم می‌شود.

$$\frac{d^2\Phi(x)}{dx^2} = -\frac{\rho(x)}{\epsilon} \quad (51.6)$$

که در آن $\Phi(x)$ پتانسیل الکتریکی (بر حسب ولت)، $\rho(x)$ چگالی بار وابسته به x (بر حسب C/m^3)، و ϵ ثابت دی‌اکتریکی ماده بین صفحات است.

مثال ۸.۶

تابع $\Phi(x)$ بین صفحات خازن شکل (۱۲.۶) با فاصله بین صفحات D متر و $\rho(x) = (\rho_0(x - D))^2$ با شرایط مرزی $\Phi(0) = 0$ و $\Phi(D) = V_0$ بدست آورید. با قرار دادن $(\rho(x))$ در معادله (۵۱.۶) خواهیم داشت:

$$\frac{d^2\Phi}{dx^2} = -\frac{\rho_0}{\epsilon}(x - D)^2 = -\frac{\rho_0}{\epsilon}(x^2 - 2Dx + D^2) \quad (52.6)$$

همانطوری که برای مدار RC در این فصل انجام دادیم، معادله (۵۲.۶) را با استفاده روش‌های عددی و کلاسیک حل کرده و جوابها را مقایسه می‌نماییم. این معادله دیفرانسیل ساده را با دوبار انتگرال‌گیری از هر دو طرف رابطه حل می‌کنیم. خواهیم داشت:

$$\Phi(x) = -\frac{\rho_0}{\epsilon} \left(\frac{1}{12}x^4 - \frac{D}{3}x^3 + \frac{D^2}{2}x^2 + \alpha x + \beta \right) \quad (53.6)$$

که در آن α و β ثابت‌های انتگرال‌گیری است و با اعمال شرایط مرزی بدست می‌آید. با اعمال اولین شرط مرزی $\Phi(0) = 0$ داریم

$$0 = -\frac{\rho_0}{\epsilon} \left(\frac{1}{12}0^4 - \frac{D}{3}0^3 + \frac{D^2}{2}0^2 + \alpha 0 + \beta \right)$$

بنابراین، $\beta = 0$ بدست می‌آید.

برای شرط مرزی دوم $\Phi(D) = V_0$ داریم

$$V_0 = -\frac{\rho_0}{\epsilon} \left(\frac{1}{12}D^4 - \frac{D}{3}D^3 + \frac{D^2}{2}D^2 + \alpha D \right)$$

و از حل آن برای α میدهد

$$\alpha = \frac{-\epsilon V_0}{D\rho_0} - \frac{D^3}{4}$$

بنابراین، حل دقیق معادله (۵۲.۶) برابر است با

$$\Phi(x) = -\frac{\rho_0}{\epsilon} \left(\frac{1}{12}x^4 - \frac{D}{3}x^3 + \frac{D^2}{2}x^2 - \frac{\epsilon V_0}{D\rho_0} + \frac{D^3}{4}x \right) \quad (۵۴.۶)$$

برای حل عددی معادله (۵۲.۶)، ما ناحیه بین دو صفحه را به N قسمت تقسیم کرده و روش تفاضلات محدود را طبق جدول (۱۲.۶) اعمال می‌کنیم. تقسیم ناحیه $0 \leq x \leq D$ به N قسمت، تعداد $N+1$ مقدار برای x و در نتیجه $N+1$ مقدار برای Φ خواهیم داشت. هدف ما تعیین ضرایب ماتریس $(N+1) \times (N+1)$ سه-قطري خواهد بود.

ما با اعمال رابطه تفاضل مرکزی مرتبه دوم از جدول (۱۲.۶) برای معادله (۵۲.۶) شروع کرده و خواهیم داشت

$$\Phi_i = \frac{\Phi_{i+1} + \Phi_{i-1} - 2\Phi_i}{h^2} = -\frac{\rho_0}{\epsilon} (x_i^2 - 2Dx_i + D^2)$$

و حل آن برای Φ_i میدهد

$$\Phi_i = \frac{1}{2}\Phi_{i+1} + \frac{1}{2}\Phi_{i-1} + \frac{\rho_0 h^2}{2\epsilon} (x_i^2 - 2Dx_i + D^2) \quad (۵۵.۶)$$

این معادله برای $i = 2, 3, 4, \dots, N$ صادق است. توجه کنید که معادله (۵۵.۶) مانند معادله (۳۲.۶) فرم کلی است. می‌توانیم عناصر ماتریس سه-قطري را چنین مشخص کنیم:

$$a_i = \frac{1}{2}, \quad b_i = \frac{1}{2}, \quad c_i = \frac{\rho_0 h^2}{2\epsilon} (x_i^2 - 2Dx_i + D^2)$$

اکنون شرایط مرزی را برای تعیین مقادیر کوشش‌های سمت چپ بالا و راست پائین ماتریس ضرایب استفاده می‌کنیم. ابتداء شرایط مرزی را بازنویسی می‌کنیم:
 $\Phi(0) = 0 \rightarrow \Phi_1 = 0$ (۵۶.۶)

$$\Phi(D) = V_0 \rightarrow \Phi_{N+1} = V_0 \quad (۵۷.۶)$$

معادله (۵۶.۶) می‌دهد

$$a_1 = b_1 = c_1 = 0$$

معادله (۵۷.۶) می‌دهد

$$a_{N+1} = 0, \quad b_{N+1} = 0, \quad c_{N+1} = V_0$$

برنامه مطلب برای حل عددی معادله (۵۱.۶) بقرار زیر است:

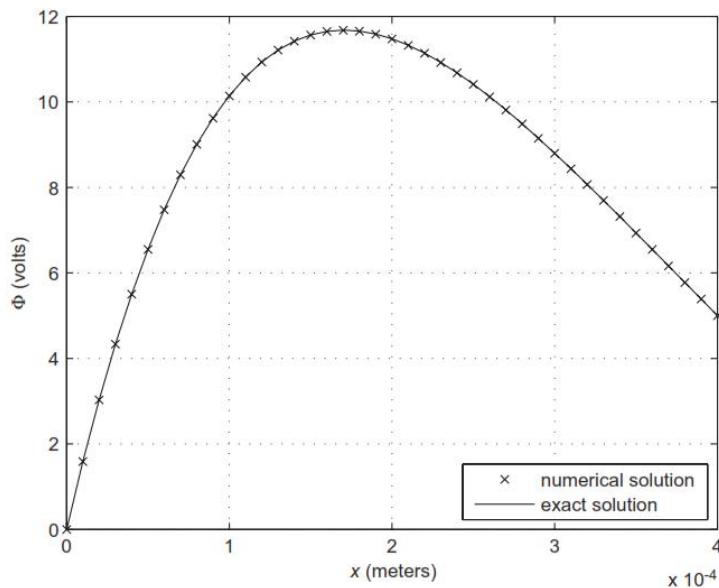
```
% Example_6_8.m
% Find the electric potential between a parallel plate
% capacitor with fixed charge density between the plates
% and known boundary conditions.
clc; clear;
N=40; % step count
D=.0004; % plate separation (meters)
h=D/N; % step size
rho_o=1e4; % coulomb/m^3
epsilon = 1.04e-12; % dielectric between the plates
Vo=5; % voltage across plates
```

```
% First, calculate all values of x. Also calculate
% the exact solution.
x = 0:h:D;
% Calculate exact solution (Equation 6.54)
phi_exact = (-rho_o/epsilon)*(x.^4/12 -(D/3)*x.^3 ...
+ (D^2/2)*x.^2 -((epsilon*Vo/(D*rho_o))+(D^3/4))*x);
% Next, solve numerically using a tri-diagonal matrix.
% Define 'a','b',and 'c'vectors:
a = .5 * ones(1,N+1);
b = .5 * ones(1,N+1);
c = (rho_o*h^2/(2*epsilon))*( x.^2 -2*D*x +D^2);
% Set boundary conditions for 'a','b',and 'c'vectors
a(1)=0; b(1)=0; c(1)=0;
a(N+1)=0; b(N+1)=0; c(N+1)=Vo;
% Solve tri-diagonal matrix by calculating "d" and
% "e" vectors
d(1)=c(1);
e(1)=a(1);
for i=2:N+1
d(i) = (c(i) + b(i)*d(i-1)) / (1 -b(i)*e(i-1));
e(i) = a(i) / (1 -b(i)*e(i-1));
end
% calculate phi based on d and e,also calculate
% global error
phi(N+1)=d(N+1);
for i=N:-1:1
phi(i) = d(i) + e(i)*phi(i+1);
end
% Plot numerical and exact solutions for phi:
plot(x,phi,'x',x,phi_exact);
xlabel('{\it x} (meters)'), ylabel('Phi (volts)');
grid;
legend('numerical solution','exact solution',...
'Location','SouthEast');
% Find worst-case global error:
error = abs( (phi -phi_exact) ./ phi_exact );
fprintf('Max global error for %d steps: %5.3f%\n',...
N,100*max(error));
%-----
```

نتایج برنامه

Maximum global error for 40 steps: 0.029%

نمودار متلب Φ بر حسب تابعی از x در شکل (۱۳.۶) نشان داده شده است.



شکل ۱۳.۶: نمودار تابع پتانسیل الکتریکی $\Phi(x)$ بین صفحات شامل حل دقیق و روش تفاضل محدود برای $N = 40$.

پروژه‌ها

۱.۶ پروژه

مدار RC شکل (۴.۶) را برای $v_C(t)$ حل کنید. ولتاژ تغذیه بصورت زیر است:

$$V_D(t) = \begin{cases} V_0 \sin \omega t & t \geq 0 \\ 0 & t < 0 \end{cases}$$

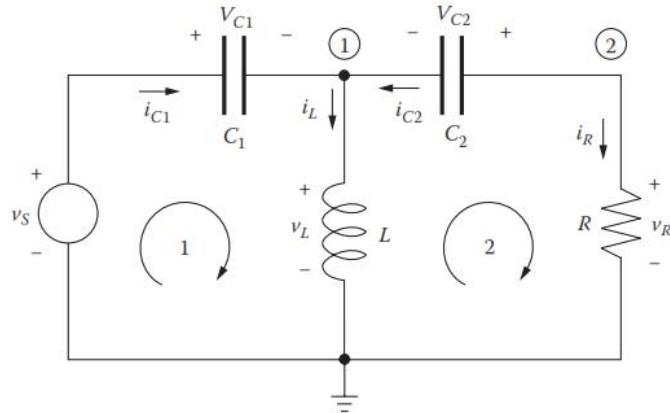
شرط اولیه را $v_C(0) = 0$ فرض کرده و مقادیر پارامترها $C = 1 \mu F$, $R = 1 k\Omega$ و $V_0 = 10 V$, $2000 rad/sec$ هستند.

۱. جواب دقیق $v_C(t)$ را با فرض جواب ویژه بصورت $v_{C,P}(t) = \alpha \sin \omega t + \beta \cos \omega t$ و استفاده از معادله (۱۴.۶) برای جواب معادله همگن که بصورت $v_{C,H} = \gamma e^{-t/\tau}$ است بدست آورید. ثابت‌های α , β , و γ را با قرار دادن جواب در معادله دیفرانسیل حاکم بر مدار و اعمال شرط اولیه بدست آورید.
۲. از آلگوریتم اولر برای حل عددی $v_C(t)$ استفاده کرده و در بازه $t = [0, 5] ms$ با قدمهای $0.1 ms$ آنرا حل کنید.
۳. از آلگوریتم اولر اصلاح شده برای حل عددی $v_C(t)$ استفاده کرده و در همان بازه بخش دوم آنرا حل کنید.
۴. نمودار $v_C(t)$ را بر حسب t برای هردو روش اولر و روش حل دقیق رسم کنید. در روی نمودار دیگری $v_C(t)$ را بر حسب t برای هردو روش اولر اصلاح شده و روش حل دقیق رسم کنید.

۵. اندازه قدمها را در حل عددی افزایش داده تا ناپایداری را شروع شود. یک قاعده سرانگشتی برای این که حل عددی این سیستم مرتبه اول با ورودی سینوزوئیدال پایدار باشد چیست؟

۲.۶ پروژه

شکل (۱۴.۶) یک مدار مرتبه سوم RLCC را نشان میدهد. برای اجرای تحلیل گذرائی حوزه زمان، ما مدار را با استفاده روش زیر به معادلات دیفرانسیل تبدیل می‌کنیم: ابتداء، قانون جریان



شکل ۱۴.۶: مدار مرتبه سوم RLCC شامل دو خازن و یک سیمپیج.

$$\text{کیرشهف را برای هر گره در مدار می‌نویسیم:} \\ i_{C1} + i_{C2} - i_L = 0 \quad \text{در گره شماره یک:} \quad (۵۸.۶)$$

$$i_R + i_{C2} = 0 \quad \text{در گره شماره دو:} \quad (۵۹.۶)$$

$$\text{ثانیاً، قانون ولتاژ کیرشهف را برای دو حلقه در مدار می‌نویسیم:} \\ v_s - v_{C1} - v_L = 0 \quad \text{در حلقه شماره یک:} \quad (۶۰.۶)$$

$$v_L + v_{C2} - v_R = 0 \quad \text{در حلقه شماره دو:} \quad (۶۱.۶)$$

$$\text{ثالثاً، روابط بین ولتاژ و جریان را برای هر عنصر مدار می‌نویسیم:} \\ i_{C1} = C_1 v'_{C1} \quad : C_1 \quad \text{برای:} \quad (۶۲.۶)$$

$$i_{C2} = C_2 v'_{C2} \quad : C_2 \quad \text{برای:} \quad (۶۳.۶)$$

$$v_L = L i'_L \quad : L \quad \text{برای:} \quad (۶۴.۶)$$

$$v_R = i_R R \quad : R \quad \text{برای:} \quad (۶۵.۶)$$

ما سه جمله مشتق دار (v_{C1}, v_{C2}, i_R) را بعنوان متغیرهای حالت برای این مسئله انتخاب می‌کنیم.

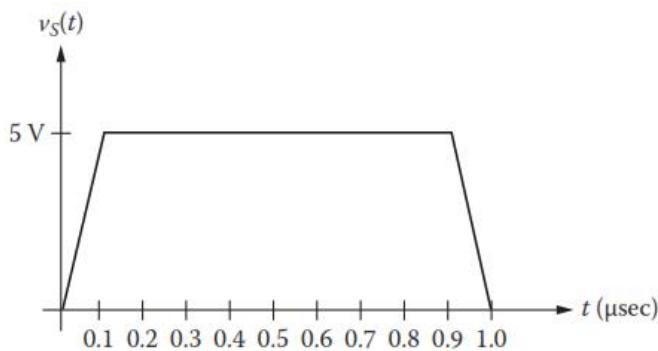
۱. معادلات (۶۶.۶)–(۶۷.۶) بصورت استاندار زیر مرتب می‌کنیم:

$$\frac{dv_{C1}}{dt} = f(v_{C1}, v_{C2}, i_L, v_S(t)) \quad (66.6)$$

$$\frac{dv_{C2}}{dt} = g(v_{C1}, v_{C2}, i_L, v_S(t)) \quad (67.6)$$

$$\frac{di_L}{dt} = h(v_{C1}, v_{C2}, i_L, v_S(t)) \quad (68.6)$$

که در آن f , g , و h شامل پارامترهای L , C_2 , C_1 , و R هستند.



شکل ۱۵.۶: پالس ورودی مدار RLCC

۲. معادله (۶۶.۶)–(۶۸.۶) با الگوریتم رانج-کوتا برای پارامترهای زیر حل کنید: $C_1 = 1 \mu F$, $C_2 = 0.001 \mu F$, $L = 0.01 mH$, $R = 100 k\Omega$, بازه زمانی $0 \leq t \leq 10 \mu s$ در نظر گیرید. فرض کنید $v_s(t) = 5 V$ یک پالس است که در $t = 0$ شروع و با زمان صعود $0.1 \mu s$ پالس تا زمان $0.8 \mu s$ روشن باقی مانده و با زمان $0.1 \mu s$ شروع خاموشی می‌گراید. این در شکل (۱۵.۶) نشان داده شده است. شرایط اولیه بقرار زیر است:

$$v_{C1}(0) = 0, \quad v_{C2}(0) = 0, \quad i_L(0) = 0$$

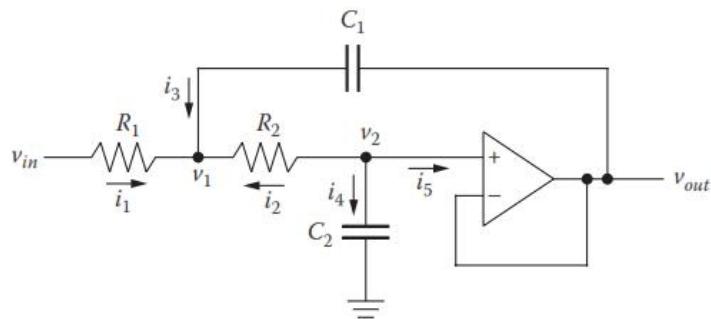
۳. در نمودارهای جدا از یکدیگر $i_L(t)$, $v_{C2}(t)$, $v_{C1}(t)$ و $v_S(t)$ را بر حسب زمان رسم کنید.

۳.۶ پروژه

مدار سالن-کی ۲۳ شکل (۱۶.۶) غالباً عنوان یک فیلتر مرتبه دوم بکار می‌رود. با وجودی که چندین راه برای مدل کردن این مدار هست، ما آن را مستقیماً در حوزه زمان با استفاده از معادلات دیفرانسیل تحلیل می‌کنیم. اگر فرض کنیم که آپ امپ ایده‌آل است، در این صورت $v_2 = v_{out}$ و $i_5 = 0$ است. با اعمال قانون جریان کیرشهف در گره‌های با نماد v_1 و v_2 و استفاده از روابط بین ولتاژ و جریان مقاومت‌ها، خواهیم داشت:

$$i_1 + i_2 + i_3 = 0, \quad \frac{v_{in} - v_1}{R_1} + \frac{v_2 - v_1}{R_2} + C_1 \frac{d(v_{out} - v_1)}{dt} = 0 \quad : v_1$$

$$i_2 + i_4 = 0, \quad \frac{v_2 - v_1}{R_2} + C_2 \frac{dv_2}{dt} = 0 \quad : v_2$$



شکل ۱۶.۶: مدار سالن-کی

با مرتب کردن سیستم دو معادله دیفرانسیل معمولی مرتبه اول و با قرار دادن $v_2 = v_{out}$ خواهیم داشت

$$\frac{dv_{out}}{dt} = \frac{-1}{R_2 C_2} v_{out} + \frac{1}{R_2 C_2} v_1 \quad (۶۹.۶)$$

$$\frac{dv_1}{dt} = \frac{1}{R_2 C_1} - \frac{1}{R_2 C_2} v_{out} + \frac{1}{R_2 C_2} - \frac{1}{R_1 C_1} - \frac{1}{R_2 C_1} v_1 + \frac{1}{R_1 C_1} v_{in} \quad (۷۰.۶)$$

۱. با استفاده از تابع `ode45` متلب v_{out} و v_1 را بدست آورید. فرض کنید که ورودی مدار v_{in} یک ولتاژ تک پله‌ای که از صفر بیک در زمان $t = 0^+$ تغییر می‌کند. برای عناصر مدار پارامترهای زیر را در نظر گیرید

$$R_1 = 5000, \quad R_2 = 5000\Omega, \quad C_1 = 2200 \text{ pF}, \quad C_2 = 1100 \text{ pF}$$

بازه زمانی را $v_{out}(0) = v_1(0) = 0$ و $t = [0, 100\mu\text{s}]$ فرض کنید.

۲. پاسخ ضربه‌ای مدار را ابتدا با تولید تابع `pulse(t)` که مقادیر زیر را برمی‌گرداند درست کنید:

$$pulse(t) = \begin{cases} 10^6 & \text{برای } 0 < t < 10^{-6} \\ 0 & \text{در غیر این صورت} \end{cases}$$

سپس، با استفاده از تابع `ode45` متلب v_{out} و v_1 را بدست آورید. در اینجا $vi = pulse(t)$ است. مقادیر عناصر، بازه زمانی، و شرایط اولیه مانند بخش اول است.

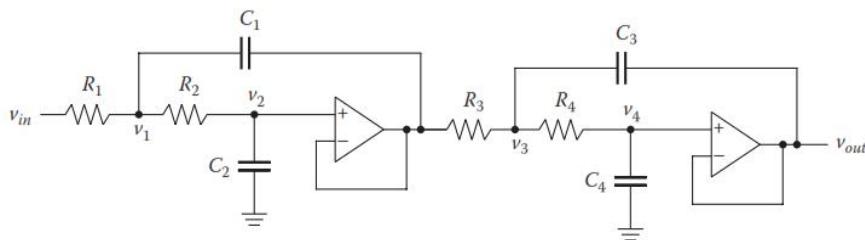
۳. پاسخ پله واحد (از بخش اول) و پاسخ ضربه‌ای (از بخش دوم) را روی یک محور رسم کنید. بین این دو چه رابطه‌ای وجود دارد؟

۴. مدار سالن-کی در مقابل مدار مرتبه دوم معمولی RLC که در پیوست (الف) آورده شده است چیست؟

۴.۶ پروژه

با بستن آبشاری مدار سالن-کی می‌توانیم فیلترهایی با مرتبه بالاتر تولید کنیم. شکل (۱۷.۶) یک نمونه از فیلتر مرتبه چهارم را نشان می‌دهد.

با استفاده از معادلات (۶۹.۶) و (۷۰.۶)، یک دستگاه معادلات دیفرانسیل مرتبه چهارم برای تشریح مدار سالن-کی مرتبه چهارم بنویسید. بخاطر داشته باشید که طبقه اول برابر v_{in} طبقه دوم است.



شکل ۱۷.۶: فیلتر مرتبه چهارم با استفاده از مدارهای سالن-کی

- با استفاده از تابع `ode45` متلب v_{out} بدست آورید. فرض کنید که ورودی مدار v_{in} یک ولتاژ پله‌واحد است که در لحظه $t = 0^+$ از $0 V$ به $1 V$ تغییر می‌یابد. مقادیر عناصر مدار را بقرار زیر فرض کنید.

$$R_1 = R_2 = R_3 = 5000\Omega, \quad C_1 = 18000 \text{ pF}, \\ C_2 = 15000 \text{ pF}, \quad C_3 = 4.7\mu\text{F}, \quad C_4 = 10 \text{ pF}$$

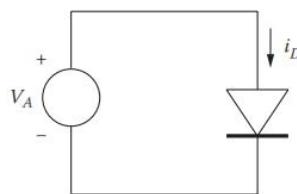
- بازه زمانی را s و شرایط اولیه $v_o(0) = v_1(0) = v_2(0) = v_3(0) = 0$ را در نظر گیرید.

- ولتاژهای t را بر حسب زمان t روی یک محور رسم کنید.

۵.۶ پروژه

یک مسئله مقدار مرزی معمولی در مهندسی برق حل پتانسیل الکتریکی در ناحیه تخلیه یک اتصال PN است. اتصال PN در اثر مجاورت دو ناحیه هم نیمه‌هادی (مثلاً سیلیکن) که مخالف یکدیگر ناخالص شده‌اند تشکیل می‌شود. و این برای تولید ادوات الکترونیکی از قبیل ترانزیستور دوقطبی، ماسفت^{۲۴} و جیفت^{۲۵} $JFET$ بکار می‌رود. در اینجا ما ساده‌ترین قطعه یعنی دیود PN را تحلیل می‌کنیم.

شکل (۱۸.۶) نمایش مداری یک دیود که ولتاژ V_A بدان اعمال شده و جریان i_D از آن می‌گذرد را نشان میدهد. شکل (۱۹.۶) نمایش فیزیکی دیود را که از اتصال دو نیمه‌هادی نوع p (که جریان



شکل ۱۸.۶: مدار دیود

اصولاً حفره‌ها^{۲۶} حامل‌هائی با چگالی N_A حامل بر سانتی متر مکعب هستند) و نیمه‌هادی نوع n (که اصولاً جریان الکترون‌ها با چگالی N_D حامل بر سانتی متر مکعب هستند) را نشان میدهد. در ناحیه‌ای که فوراً اطراف اتصال را تشکیل میدهد، در شکل مفصلًاً نشان داده شده است، حامل‌های بار متحرک ناحیه نیمه‌هادی نوع p و نوع n با یکدیگر ترکیب شده و ناحیه تخلیه^{۲۷}

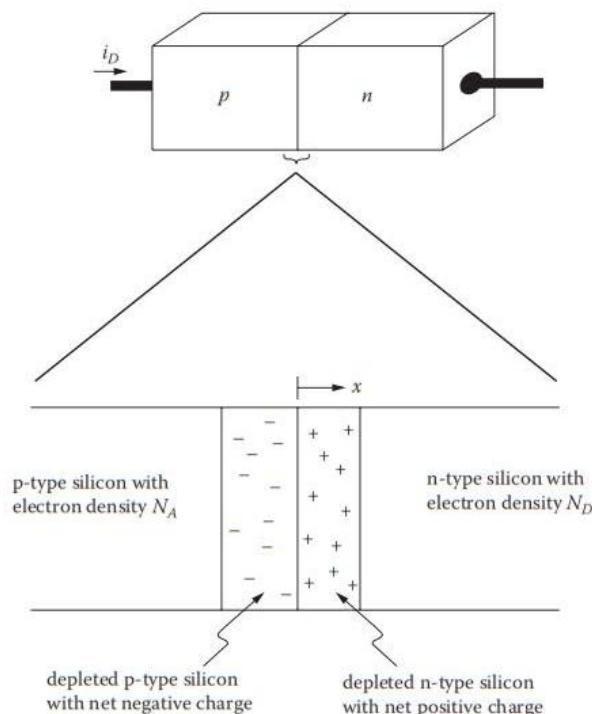
^{۲۴}metal oxide semiconductor field effect transistors (MOSFET)

^{۲۵}junction field effect transistor (JFET)

^{۲۶}holes

^{۲۷}depletion region

را تشکیل میدهند. چگالی بار در ناحیه تخلیه بستگی به شکل ناخالصی^{۲۰.۶}، دارد و نمونه‌ای از شکل ناخالصی در شکل (۲۰.۶) نشان داده شده است. در این شکل، ما دو تقریب بکار برده‌ایم: (الف) آنکه ترازهای (سطوح) ناخالصی سیلیکن از N_A به N_D در اتصال سریعاً انجام شود (یعنی در $x = 0$ ، و (ب) آنکه ناحیه تخلیه در $-x_P$ - روی ضلع چپ و در x_N روی ضلع راست سریعاً خاتمه یابد. بنابراین، چگالی بارها را می‌توان بصورت خط مستقیم تقریب زد.



شکل ۱۹.۶: بستن کامل ناحیه تخلیه یک اتصال PN

$$\rho_P = -qN_A \quad -x_P < x < 0 \quad \text{برای} \quad (71.6)$$

$$\rho_N = qN_D \quad 0 < x < x_N \quad \text{برای} \quad (72.6)$$

که در آن ρ_P و ρ_N چگالی بارها بترتیب در ناحیه نوع p و نوع n و نوع q واحد بار الکتریکی $(1.6 \times 10^{-19} C)$ است. (توجه کنید که چگالی سمت P ناحیه تخلیه منفی است زیرا از حامل‌های مثبت تخلیه شده است. بهمین‌نحو، طرف N دارای چگالی بار مثبت است).

معادله حاکم برای پتانسیل Φ در ناحیه تخلیه نوع p برابر است با:

$$\frac{d^2\Phi}{dx^2} = \frac{qN_A}{\epsilon} \quad (73.6)$$

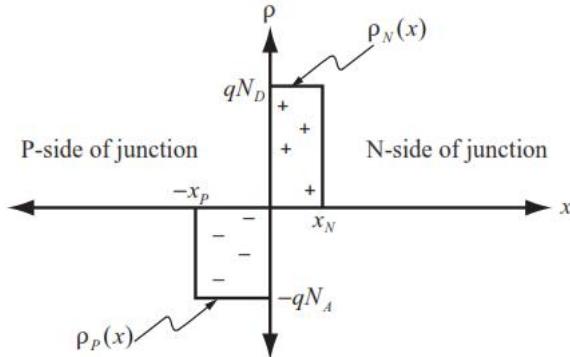
و معادله حاکم برای پتانسیل Φ در ناحیه تخلیه نوع n برابر است با:

$$\frac{d^2\Phi}{dx^2} = -\frac{qN_D}{\epsilon} \quad (74.6)$$

^{۲۰}doping profile

ما $x = 0$ را درست در نقطه گذر P به N قرار داده و پتانسیل را بطور دلخواه $V = 0$ در فرض می‌کنیم، یعنی $x = 0$

$$\Phi(0) = 0 \quad (75.6)$$



شکل ۲۰.۶: چگالی بار $\rho_P(x)$ و $\rho_N(x)$ در اتصال PN

شرط مرزی دوم این است که میدان الکتریکی باید در ناحیه خنثی خارج از ناحیه تخلیه صفر باشد، زیرا هیچ‌گونه بار خالصی در آن ناحیه وجود ندارد. چون $E = -\frac{d\Phi}{dx}$ است، این شرط میگوید که

$$\Phi = 0 \quad \text{در} \quad x = -x_P \quad (76.6)$$

$$\Phi = 0 \quad \text{در} \quad x = x_N \quad (77.6)$$

فرض کنید که میتوانیم اتصال PN را بصورت خازن صفحات موازی که صفحات در $x = x_N$ و $x = -x_P$ قرار دارند مدل کنیم. با استفاده از روشی که در مثال ۸.۶ بکار برده شد، پتانسیل الکتریکی در ناحیه تخلیه را محاسبه کنید. برای این منظور پارامترهای زیر را در نظر گیرید:

$$q = 1.6 \times 10^{-19} C, \quad N_D = 10^{16} carrier/cm^3, \quad N_A = 0.3 \times 10^{16} carrier/cm^3$$

$$x_N = 0.15 \mu m, \quad x_P = 0.5 \mu m, \quad \Delta x = 0.005 \mu m, \quad \epsilon = 1.04 \times 10^{-12} F/m$$

۱. در ناحیه نوع p ، پتانسیل Φ را در فواصل $0 \leq x \leq -x_P$ با شرایط مرزی $\Phi(-x_P) = 0$ و $\Phi(0) = 0$ حل کرده و با استفاده از معادلات تفاضلات محدود ماتریس سه-قطري را تعریف کرده، سپس ماتریس را حل کنید.

۲. برای ناحیه نوع n پتانسیل Φ را در فواصل $0 \leq x \leq x_N$ با شرایط مرزی $\Phi(0) = 0$ و $\Phi(x_N) = 0$ حل کرده و با استفاده از روشی که در بخش اول کفته شد جواب عددی را بدست آورید.

۳. نتایج خود را برای هردو ناحیه نوع p و n روی یک نمودار در فواصل $-x_P \leq x \leq x_N$ رسم کنید.

• پروژه ۶.۶
برای اتصال PN در مسئله ۵.۶، ما بطور دلخواه نقطه زمین را $\Phi = 0$ درست در نقطه مرزی

اتصال P و N قرار دادیم. فرض کنید که ما نقطه زمین را در مرز P به هوا قرار دهیم، یعنی معادله برای ناحیه p خواهد شد $\Phi(-x_P) = 0$

$$\frac{d^2\Phi}{dx^2} = \frac{qN_A}{\epsilon} \quad (78.6)$$

$$\Phi(-x_P) = 0 \quad (79.6)$$

$$\Phi(x_N) = 0 \quad (80.6)$$

و برای ناحیه نوع n

$$\frac{d^2\Phi}{dx^2} = -\frac{qN_D}{\epsilon} \quad (81.6)$$

در مقطع دو ناحیه، جائی که $x = 0$

$$\Phi(0^-) = \Phi(0^+) \quad (82.6)$$

$$\frac{d\Phi}{dx}(0^-) = \frac{d\Phi}{dx}(0^+) \quad (83.6)$$

۱. جواب دقیق را برای معادلات (۷۸.۶)-(۸۳.۶) با انتگرال‌گیری مستقیم بدست آورید.

۲. برای معادلات (۷۸.۶)-(۸۳.۶) با استفاده از تابع `ode45` متلب، جواب عددی را بدست آورید. بفرض آنکه:

$$N_D = 10^{16} \text{ carrier/cm}^3, \quad N_A = 0.3 \times 10^{16} \text{ carrier/cm}^3, \quad \epsilon = 1.04 \times 10^{-12} F/m$$

$$\text{در ناحیه } -x_P \leq x \leq x_N$$

$$x_P = 0.5 \mu m, \quad x_N = 0.15 \mu m, \quad \Delta x = 0.005 \mu m$$

۳. جواب دقیق و جواب عددی `ode45` را روی یک نمودار رسم کنید.

۴. خطای جهانی در جواب عددی `ode45` را بدست آورده و چاپ کنید.

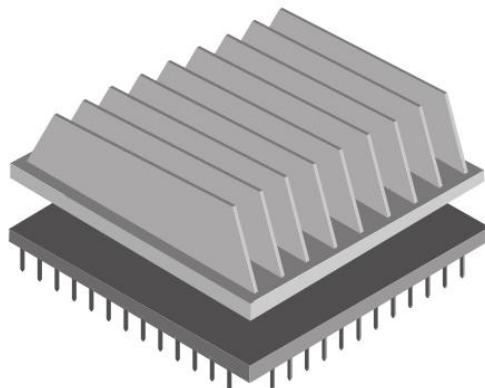
۷.۶ پروژه

یک گرماییر ^{۲۹} اغلب به عناصر مدارات الکترونیکی متصل می‌کنند تا حرارت مازاد تولید شده بوسیله ادوای را برای جلوگیری از گرمایش بیش از حد، تلف کند (شکل ۲۱.۶). شکل و شما میل پره‌های یک گرماییر در شکل (۲۲.۶) نشان داده شده است. در اینجا یک پره ذوزنقه‌ای شکل هست که عرض قاعده W_1 ، عرض بالای W_2 ، ارتفاع L_1 ، و عمق آن D است. محورهای مختصات x و y در شکل ترسیم شده است. توزیع دما در پره با تحلیل یک بعدی معادله حرارت در جامدات میسر است. تحلیل شامل یک رابطه تجربی است که ضریب انتقال حرارت هموفت h میباشد. این رابطه تجربی برای جدا سازی مسئله هدایت حرارتی در جامدات از مسئله انتقال حرارت در مایعات اطراف را در بر می‌گیرد. ضریب انتقال حرارت هموفت h با آزمایش یا روش عددی/تحلیلی بدست می‌آید. معادلات حاکم برای پره چنین است:

$$\frac{d}{dx} \left[A \frac{dT}{dx} \right] = \frac{hP}{k} (T - T_\infty) \quad (84.6)$$

یا

$$\frac{dA}{dx} \frac{dT}{dx} + A \frac{d^2T}{dx^2} = \frac{hP}{k} (T - T_\infty) \quad (85.6)$$



شکل ۲۱.۶: نمونه‌ای از پره‌های گرم‌گیر *heat sink* که در خنک‌کننده‌ای قطعات الکترونیکی از قبیل میکروپروسسورها بکار می‌رود.

که T دمای در موقعیت x پره، T_∞ دمای هوای محیط اطراف، A سطح مقطع پره، h ضریب انتقال حرارتی همرفت، k هدایت حرارتی ماده پره، و P محیط هندسی آن است. ما با نوشتن معادله برای سطح و محیط هندسی بر حسب اندازه‌گیری پره شروع می‌کنیم:

$$A = yD$$

$$P = 2y + 2D$$

با شکل هندسی شبیه مثلث، میتوانیم $\frac{y}{W_1} = \frac{L-x}{L}$ را نوشه و بنابراین سطح و محیط بر حسب x خواهد بود

$$A(x) = \frac{W_1 D}{L} (L - x)$$

$$P(x) = \frac{2W_1 D}{L} (L - x) + 2D$$

معادله دیفرانسیل حاکم بر مسئله خواهد شد

$$\frac{W_1 D}{L} (L - x) \frac{d^2 T}{dx^2} - \frac{W_1 D}{L} \frac{dT}{dx} = \frac{h}{k} \left[\frac{2W_1}{L} (L - x) + 2D \right] (T - T_\infty) \quad (86.6)$$

اولین شرط مرزی در سطح قطعه دستگاه دمای T_W است و میدانیم

$$T(0) = T_W \quad (87.6)$$

برای تعیین شرط مرزی دوم، ما از مفهوم اصل بقاء انرژی استفاده می‌کنیم، یعنی

نرخی که حرارت بوسیله انتقال هدایتی پره را ترک می‌کند در ($x = L_1$) بر واحد سطح پره

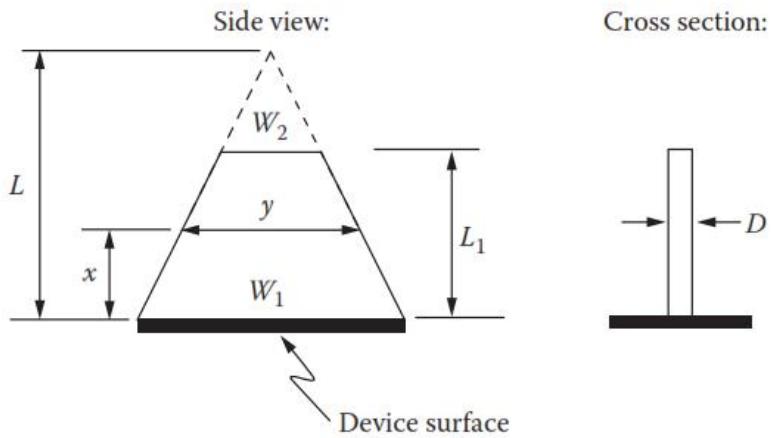
نرخی که حرارت با انتقال همرفتی بر واحد سطح پره دور می‌شود

این بیان را می‌توان از نظر ریاضی چنین نوشت:

$$-k \frac{dT}{dx}(L_1) = h[T(L_1) - T_\infty] \quad (88.6)$$

می‌خواهیم این مسئله را بصورت عددی با استفاده از روش تفاضلات محدود حل کنیم. ابتدا، x را به N قسمت تقسیم کرده و خواهیم داشت $x_1, x_2, x_3, \dots, x_{N+1}$. دمای در x_i را T_i در نظر

^{۲۹}heat sink



شکل ۲۲.۶: منظره مقطع و پهلوی یک پره گرمگیر با ارتفاع ذوزنقه L , قاعده پائین W_1 , قاعده بالا W_2 , و عمق D . ذوزنقه از یک بریدن یک مثلث به ارتفاع L حاصل می‌شود.

می‌گیریم. فرمولهای تفاضلات محدود برای $\frac{dT}{dx}$ و $\frac{d^2T}{dx^2}$ (به جدول ۱۲.۶ مراجعه کنید) برابر است با:

$$\frac{d^2T}{dx^2}(x_i) = \frac{T_{i+1} + T_{i-1} - 2T_i}{(\Delta x)^2}$$

$$\frac{dT}{dx}(x_i) = \frac{T_{i+1} - T_i}{\Delta x}$$

فرم تفاضل محدود معادله (۸۶.۶) خواهد شد:

$$\frac{W_1 D}{L} (L - x_i) \left[\frac{T_{i+1} + T_{i-1} - 2T_i}{(\Delta x)^2} \right] - \frac{W_1 D}{L} \left[\frac{T_{i+1} - T_i}{\Delta x} \right] = \frac{h}{k} \left[\frac{2W_1}{L} (L - x_i) + 2D \right] (T_i - T_\infty)$$

از حل این رابطه برای T_i خواهیم داشت

$$\begin{aligned} T_i &= \left\{ \frac{2W_1 D}{L} (L - x_i) - \frac{W_1 D}{L} (\Delta x) + \frac{h(\Delta x)^2}{k} \left[\frac{2W_1 D}{L} (L - x_i) + 2D \right] \right\}^{-1} \\ &\times \left\{ \left[\frac{W_1 D}{L} (L - x_i) - \frac{W_1 D}{L} (\Delta x) \right] T_{i+1} + \frac{W_1 D}{L} (L - x_i) T_{i-1} \right. \\ &\left. + \frac{h(\Delta x)^2}{k} \left[\frac{2W_1 D}{L} (L - x_i) + 2D \right] T_\infty \right\} \end{aligned} \quad (۸۹.۶)$$

معادله (۸۹.۶) برای $i = 2, 3, \dots, N$ صادق است. فرم تفاضل محدود برای معادله (۸۸.۶) برابر است با

$$-k \frac{T_{N+1} - T_N}{\Delta x} = h [T_{N+1} - T_\infty]$$

و با حل این برای T_{N+1} میدهد

$$T_{N+1} = \frac{1}{1 + \frac{h \Delta x}{k}} \left\{ T_N + \frac{h \Delta x}{k} T_\infty \right\}$$

$$T_1 = T_w \quad \text{همچنین،}$$

تمام انتقال حرارت از پره به هوا اطراف از طریق قاعده پره انجام می‌شود. بنابراین، نرخ اتلاف حرارتی Q از پره خواهد بود

$$Q = -\frac{kA_1}{\Delta x}(T_2 - T_1)$$

که در آن A_1 سطح مقطع در قاعده پره، یعنی $x = 0^+$ است.

با استفاده از روشی که در بخش ۱۱.۶ برای دستگاه معادلات سه قطری تشریح شد، برنامه‌ای بزیان متلب نوشته تا توزیع دما و نرخ اتلاف حرارت را در یک پره بدست آورد. از پارامترهای زیر در برنامه خود استفاده کنید:

$$T_W = 200^\circ C, \quad T_\infty = 40^\circ C, \quad k = 204 W/m - K, \quad h = 60 W/m^2 - K,$$

$$N = 50, \quad W_1 = 5 \text{ cm}, \quad L = 20 \text{ cm}, \quad L_1 = 5 \text{ cm}, \quad D = 0.2 \text{ cm}$$

خروجی برنامه شما باید مقادیر T در مقابل x در هر 0.1 cm و نمودار T در مقابل x باشد.

فصل ۷

تبدیل لاپلاس

۱.۷ مقدمه

روش تبدیل شامل اعمال یک عملگر ریاضی به معادلات یک مسئله، حل آن در حوزه تبدیل (که معمولاً^۱ ساده‌تر است)، و سپس عکس تبدیل برای تعیین جواب در حوزه مورد نظر است. **تبدیل لاپلاس**^۱ اغلب برای حل معادلت دیفرانسیل معمولی و جزئی بکار می‌رود. این روش معادله دیفرانسیل معمولی را بیک معادله جبری تقلیل داده، بطوری که با عملیاتی روی آن می‌توان بسهولت از آن تبدیل معکوس را بدست آورد. در نظریه مدار، هر مدار شامل خازنهای سیم پیچهایی است که شامل لاقل یک رابطه مشتق بوده، و مدارهای بزرگ ممکن است خیلی بیشتر داشته باشند. در فصل ششم، مدارها را مستقیماً با انتگرال‌گیری از معادلات دیفرانسیل در حوزه زمان در متلب حل کردیم. اکنون رویکرد تبدیل لاپلاس [۱۳-۲۴] را تشریح می‌کنیم؛ معادلات حوزه زمان را به حوزه لاپلاس تبدیل کرده، از نظر جبری آنرا حل کرده، و سپس تبدیل معکوس لاپلاس گرفته و در حوزه زمان جواب نهایی را بدست خواهیم آورد. گاهی اوقات، قدم آخر را نیز اجرا نکرده و همانجا مشکل خود را حل می‌کنیم زیرا حوزه لاپلاس بقدر کافی جواب مسئله مورد نظر ما را دربردارد.

۲.۷ تبدیل لاپلاس و تبدیل معکوس

فرض کنید $f(t)$ یک تابع علیٰ باشد بدین گونه که آن برای مقادیر $t \geq 0$ تعریف شده و در غیر این صورت صفر، یعنی $f(t) = 0$ برای $t < 0$ است. **تبدیل لاپلاس** یک طرفه بصورت زیر تعریف می‌شود

$$\mathcal{L}[f(t)] = F(s) = \int_0^{\infty} e^{-st} f(t) dt \quad (1.7)$$

تابع $F(s)$ را **تبدیل لاپلاس** $f(t)$ می‌نامند. علامت \mathcal{L} برای عملگر تبدیل است. بر طبق قرارداد، ما حروف کوچک را برای حوزه زمان و حروف بزرگ را برای حوزه لاپلاس (یا "حوزه s ") بکار می‌برند.

عکس تبدیل لاپلاس $F(s)$ را تابع $f(t)$ تعریف می‌کنند، به گونه‌ای که

$$\mathcal{L}^{-1}[F(s)] = f(t) \quad (2.7)$$

جداولی برای برخی توابع $f(t)$ و $F(s)$ مربوطه تولید کرده‌اند (به جدول در انتهای این فصل مراجعه کنید). اغلب، لازم است تا تبدیل لاپلاس را بیک فرم یا چندین فرم که در جدول تبدیلات لاپلاس مورد نیاز است مرتب و تنظیم کنیم. اکنون تبدیل لاپلاس چند تابع عمومی را بدست می‌آوریم.

¹Laplace Transforms

۱.۲.۷ تبدیل لاپلاس تابع پله واحد

ما تابع پله واحد $u(t)$ را از نظر ریاضی چنین تعریف می‌کنیم

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

در این صورت

$$\begin{aligned} \mathcal{L}[u(t)] &= \int_0^\infty (1)e^{-st}dt \\ &= -\frac{e^{-st}}{s} \Big|_0^\infty \\ &= \frac{1}{s} \end{aligned} \quad (۳.۷)$$

۲.۲.۷ تابع نمائی

فرض کنید

$$f(t) = e^{at}$$

که در آن a مقدار ثابتی است. در این صورت،

$$\begin{aligned} \mathcal{L}(e^{at}) &= \int_0^\infty e^{at}e^{-st}dt \\ &= \int_0^\infty e^{-(s-a)t}dt \\ &= -\frac{e^{-(s-a)t}}{(s-a)} \Big|_0^\infty \\ &= \frac{1}{s-a} \end{aligned} \quad (۴.۷)$$

۳.۲.۷ خطی بودن

فرض کنید

$$f(t) = Ag(t) + Bh(t)$$

که در آن $g(t)$ و $h(t)$ دو تابع و A و B مقادیر ثابتی هستند. در این صورت

$$\begin{aligned} \mathcal{L}[Ag(t) + Bh(t)] &= \int_0^\infty [Ag(t) + Bh(t)]dt \\ &= A \int_0^\infty g(t)e^{-st}dt + B \int_0^\infty h(t)e^{-st}dt \\ &= A\mathcal{L}[g(t)] + B\mathcal{L}[h(t)] \end{aligned} \quad (۵.۷)$$

بنابراین، تبدیل لاپلاس مجموع چند تابع برابر است با جمع تبدیل لاپلاس هر یک از توابع و اگر هر تابع ضریبی داشته باشد در تبدیل لاپلاس آن ضرب میشود.

۴.۲.۷ تاخیر زمانی

فرض کنید

$$f(t) = g(t - T)$$

که در آن $f(t)$ تاخیر یافته تابع $g(t)$ با زمان تاخیر T ثانیه (مسلماً مثبت است) میباشد. در این صورت،

$$\mathcal{L}[g(t - T)] = \int_T^\infty g(t - T)e^{-st}dt$$

در اینجا کرانه پائینی انتگرال را T قرار دادیم زیرا $g(t - T)$ بر فرض مسلم برای زمانهای $t < T$ صفر است. بنابراین، فرض کنید $\tau = t - T$ و در این صورت $d\tau = dt$. بنابراین،

$$\begin{aligned}\mathcal{L}[g(t - T)] &= \int_0^\infty g(\tau) e^{-(s(\tau+T))} d\tau \\ &= e^{-sT} \int_0^\infty g(\tau) e^{-s\tau} d\tau \\ &= e^{-sT} G(s)\end{aligned}\quad (6.7)$$

بنابراین، تبدیل لاپلاس یک تابع تاخیری برابر است با تبدیل تابع تاخیر نیافته ضربدر عامل تاخیر e^{-sT} .

۵.۲.۷ نمائی مختلط

فرض کنید

$$f(t) = e^{j\omega t}$$

سپس، رابطه (۴.۷) را اعمال می‌کنیم، خواهیم داشت

$$\begin{aligned}\mathcal{L}[e^{j\omega t}] &= \frac{1}{s - j\omega} \\ &= \frac{1}{s - j\omega} \times \frac{s + j\omega}{s + j\omega} \\ &= \frac{s + j\omega}{s^2 + \omega^2} \\ &= c + j \frac{\omega}{s^2 + \omega^2}\end{aligned}\quad (7.7)$$

با بکار بردن اتحاد $e^{jx} = \cos x + j \sin x$ ، ضمناً میدانیم که

$$\begin{aligned}\mathcal{L}[e^{j\omega t}] &= \mathcal{L}[\cos \omega t + j \sin \omega t] \\ &= \mathcal{L}(\cos \omega t) + j \mathcal{L}(\sin \omega t)\end{aligned}\quad (8.7)$$

با برابر قرار دادن مولفه حقیقی و موهومی معادله (۷.۷) و (۸.۷)، تبدیل لاپلاس سینوس و کسینوس را بدست خواهیم آورد:

$$\mathcal{L}(\cos \omega t) = \frac{s}{s^2 + \omega^2} \quad (9.7)$$

$$\mathcal{L}(\sin \omega t) = \frac{\omega}{s^2 + \omega^2} \quad (10.7)$$

این مطلب را می‌توان عمومیت برای حالت نمائی مختلط که هم قسمت حقیقی و موهومی دارد،

$$\begin{aligned}\mathcal{L}[e^{a+j\omega t}] &= \frac{1}{s - (a + j\omega)} \\ &= \frac{1}{(s - a) - j\omega} \\ &= \frac{s - a}{(s - a)^2 + \omega^2} + j \frac{\omega}{(s - a)^2 + \omega^2} \\ &= \mathcal{L}(e^{at} \cos \omega t) + j \mathcal{L}(e^{at} \sin \omega t)\end{aligned}$$

می‌توانیم از این چنین نتیجه گیری کنیم که ضرب تابع $e^{j\omega t}$ در $f(t) = e^{at}$ در حوزه زمان یک انتقال (شیفت) s در حوزه لاپلاس حاصل می‌شود. این را می‌توان بصورت زیر بیان کرد

$$\mathcal{L}[e^{at} f(t)] = F(s - a) \quad (11.7)$$

۶.۲.۷ توانهای t

فرض کنید،

$$f(t) = t^{n+1}$$

در این صورت

$$\mathcal{L}(t^{n+1}) = \int_0^\infty t^{n+1} e^{-st} dt$$

برای حل، از انتگرال جز به جز استفاده می‌کنیم. فرض کنید

$$dv = e^{-st} dt$$

$$du = (n+1)t^n dt$$

$$v = -\frac{e^{-st}}{s}$$

$$\int u dv = uv - \int v du$$

$$\int_0^\infty t^{n+1} e^{-st} dt = -t^{n+1} \frac{e^{-st}}{s} \Big|_0^\infty + \frac{n+1}{s} \int_0^\infty t^n e^{-st} dt = \frac{n+1}{s} \mathcal{L}(t^n) \quad (12.7)$$

(12.7)

$$\begin{aligned} \mathcal{L}(t^n) &= \frac{n}{s} \mathcal{L}(t^{n-1}) \\ \mathcal{L}(t^{n-1}) &= \frac{n-1}{s} \mathcal{L}(t^{n-2}) \\ &\vdots \\ \mathcal{L}(t^1) &= \frac{1}{s} \mathcal{L}(t^0) = \frac{1}{s} \mathcal{L}(t^0) = \frac{1}{s^2} \end{aligned} \quad (13.7)$$

۷.۲.۷ تابع ضربه

تابع دلتا^۱ توسط فیزیکدان پاول دیراک^۲ پیشنهاد و فرمول‌بندی شد و چنین تعریف می‌شود:

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & \text{در غیر این صورت} \end{cases}$$

۶

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

با وجودی که $\delta(t)$ از نظر فیزیکی در مدارات واقعی (و اغلب آنرا تقریب می‌زنند) قابل درک نیست، اما کاربردهای زیادی در نظریه مدار دارد. و تبدیل لاپلاس آن چنین است.

$$\mathcal{L}[\delta(t)] = \int_0^\infty \delta(t) e^{-st} dt = e^{-st} \Big|_0^\infty = 1$$

مثال ۱.۷:

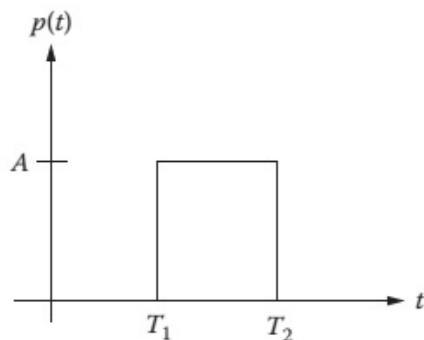
تبدیل لاپلاس پالس شکل (۱.۷) را بدست آورید. در اینجا A دامنه، T_1 زمان روشن شدن، و T_2 زمان خاموش شدن پالس است.

از نظر ریاضی می‌توانیم پالس را بصورت جمع تاخیر یافته دو تابع پله واحد به دامنه A بیان کنیم:

$$p(t) Au(t - T_1) - Au(t - T_2)$$

^۱delta function

^۲Paul Dirac



شکل ۷.۱: پالسی با دامنه A , زمان روشن شدن T_1 و زمان خاموش شدن T_2 است.

اکنون از رابطه (۳.۷) میدانیم که تبدیل لاپلاس تابع پله واحد $\frac{1}{s}$ است. همچنین از رابطه (۵.۷)، تبدیل جمع دو تابع، جمع تبدیل لاپلاس‌های دو تابع میباشد. بعلاوه، از رابطه (۶.۷) میدانیم که تبدیل لاپلاس تابع تاخیر یافته برابر حاصل ضرب $e^{-sT} e^{-sT}$ در تبدیل لاپلاس بدون تاخیر است. بنابراین تبدیل لاپلاس پالس خواهد بود

$$\begin{aligned} P(s) &= (A)(e^{-sT_1}) \frac{1}{s} - (A)(e^{-sT_2}) \frac{1}{s} \\ &= \frac{A}{s} (e^{-sT_1} - e^{-sT_2}) \end{aligned} \quad (۱۴.۷)$$

مثال ۲.۷

عكس تبدیل لاپلاس تابع زیر را بدست آورید:

$$F(s) = \frac{1 - e^{-sT}}{s^2} + \frac{2\pi T^2 e^{-sT}}{T^2 s^2 + 4\pi^2} \quad (۱۵.۷)$$

با تجزیه اولین جمله رابطه (۱۵.۷) شروع میکنیم:

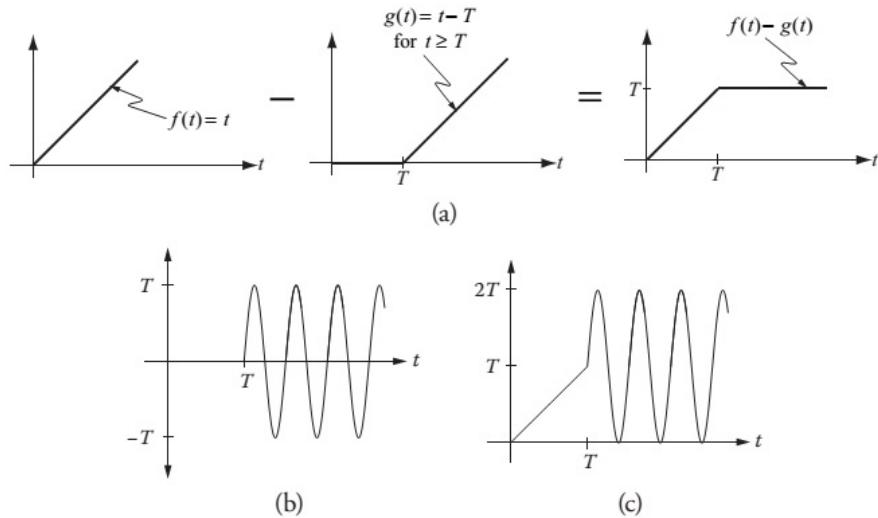
$$\frac{1 - e^{-sT}}{s^2} = \frac{1}{s^2} - e^{-sT} \frac{1}{s^2} \quad (۱۶.۷)$$

از رابطه (۱۳.۷) میدانیم که $\mathcal{L}(t) = \frac{1}{s}$ و از رابطه (۶.۷)، ضرب e^{-sT} در حوزه لاپلاس مربوط به تاخیر در حوزه زمان است. بنابراین، عکس تبدیل لاپلاس رابطه (۱۵.۷) برابر کسر تابع تک شیبی $f(t) = t$ است. و یک تابع تک شیبی تاخیر یافته. اینها در شکل (۲.۱) نمایش داده شده است.

برای جمله دوم رابطه (۱۵.۷)، ما آنرا بصورت زیر مرتب می‌کنیم:

$$\begin{aligned} \frac{2\pi T^2 e^{-sT}}{(Ts)^2 + 4\pi^2} &= T e^{-sT} \frac{2\pi T}{T^2 s^2 + (2\pi)^2} \\ &= T e^{-sT} \frac{\frac{2\pi}{T}}{s^2 + (\frac{2\pi}{T})^2} \\ &= T e^{-sT} \frac{\omega}{s^2 + \omega^2} \end{aligned} \quad (۱۷.۷)$$

با کمی جستجو، رابطه (۱۷.۷) مربوط به یک موج سینوسی است که دامنه آن T و فرکانس $\frac{2\pi}{T}$ است. آن در شکل (۲.۱)(ب) نمایش داده شده است. عکس تبدیل لاپلاس کامل در شکل (۲.۱)(ج) نشان داده شده است.



شکل ۲.۷

۳.۷ تبدیل لاپلاس مشتق

فرض کنید

$$f(t) = \frac{df(t)}{dt}$$

در این صورت

$$\mathcal{L}[f(t)] = \int_0^\infty \frac{df(t)}{dt} e^{-st} dt = \int_0^\infty e^{-st} df$$

از انتگرال جزء به جزء استفاده می‌کنیم. فرض کنید $v = e^{-st}$ و $dv = -se^{-st} dt$. بنابراین، $f = u$ و $u = e^{-st}$ است. با اعمال $\int udv = uv - \int vdu$ داشت

$$\int_0^\infty e^{-st} df = fe^{-st} \Big|_{t=0} + s \int_0^\infty fe^{-st} dt = -f(0) + s\mathcal{L}[f(t)] \quad (18.7)$$

بنابراین، تبدیل لاپلاس مشتق برابر است با تبدیل لاپلاس تابع اصلی ضرب در s منهای مقدار تابع در لحظه $t = 0$ (یعنی منهای شرط اولیه).

بهمین منوال، برای مشتق دوم،

$$\mathcal{L}[f''(t)] = \int_0^\infty f''(t) e^{-st} dt$$

$$\mathcal{L}[f''(t)] = \int_0^\infty \frac{df'(t)}{dt} e^{-st} dt = \int_0^\infty e^{-st} df'$$

فرض کنید $v = f'$ و $dv = -se^{-st} dt$. در این صورت $u = e^{-st}$ و $du = -se^{-st} dt$. با انتگرال‌گیری جزء به جزء خواهیم داشت

$$\begin{aligned} \int_0^\infty f'' e^{-st} dt &= fe^{-sT} \Big|_{t=0}^\infty + s \int_0^\infty fe^{-st} dt \\ &= -f''(0) + s\mathcal{L}[f'(t)] \end{aligned} \quad (19.7)$$

$$= -f''(0) - sf'(0) + s^2 \mathcal{L}[f(t)]$$

با مشاهده معادلات (۱۸.۷) و (۱۹.۷)، می‌توانیم الگویی برای مشتق n ام بحسب آوریم، یعنی:

$$\mathcal{L}[f^{(n)}(t)] = s^n \mathcal{L}[f(t)] - s^{n-1} f(0) - s^{n-2} f'(0) - \cdots - f^{(n)}(0) \quad (۲۰.۷)$$

۴.۷ معادلات دیفرانسیل معمولی، مسائل مقدار اولیه

معادله دیفرانسیل ناشی از مدار موازی RLC پیوست (الف) معادله (۲۰.۷) را در نظر گیرید. معادله دیفرانسیل حاکم برای جریان سیم پیچ i برابر است با

$$i'' + \frac{1}{RC} i' + \frac{1}{LC} i = \frac{I_0(t)}{LC} \quad (۲۱.۷)$$

که در آن L ضریب خودالقائی (بر حسب هانری)، R مقاومت (بر حسب اهم)، C ظرفیت (بر حسب فاراد)، و I_0 جریان منبع است.

همچنان فرض بر این است که شرایط اولیه چنین باشد،

$$i(0) = \alpha \quad \text{و} \quad i'(0) = \beta$$

ابتداء، این معادله را بصورت کلی زیر تبدیل می‌کنیم: فرض کنید $y = i(t)$ و $p = \frac{1}{LC}$ باشد. در این صورت، معادله (۲۰.۷) خواهد شد

$$y'' + py' + qy = f(t) \quad (۲۲.۷)$$

تبدیل لاپلاس هر جمله از معادله (۲۲.۷) بصورت زیر خواهد شد:

$$\mathcal{L}[y''] = s^2 Y(s) - \alpha s - \beta$$

$$\mathcal{L}[py'] = p[sY(s) - \alpha]$$

$$\mathcal{L}[qy] = qY(s)$$

بنابراین، معادله (۲۲.۷) می‌شود

$$s^2 Y(s) - \alpha s - \beta + p[sY(s) - \alpha] + qY(s) = F(s)$$

$$\rightarrow Y(s) = \frac{(s+p)\alpha + \beta + F(s)}{s^2 + ps + q}$$

فرض کنید

$$H(s) = \frac{1}{s^2 + ps + q}$$

در این صورت

$$Y(s) = [(s+p)\alpha + \beta]H(s) + F(s)H(s) \quad (۲۳.۷)$$

با انجام بسط کسرها برای رابطه (۲۳.۷) و استفاده از جداول تبدیل لاپلاس، می‌توانیم $y(t)$ را با پیدا کردن تبدیل لاپلاس معکوس ($\mathcal{L}^{-1}[Y(s)] = y(t)$) تعیین کنیم، یعنی

:۳.۷ مثال

معادله دیفرانسیل زیر داده شده است (ضریب میرائی ندارد و تغذیه تابع میراشهونده است)، جواب ($y(t)$) را بحسب آورید.

$$y'' + y = 5e^{-t}, \quad y(0) = 2, \quad y'(0) = 0 \quad (۲۴.۷)$$

این مسئله با فرم کلی معادله (۲۲.۷) با $f = 5e^{-t}$ ، $q = 1$ ، $p = 0$ جور می‌شود. بنابراین،

$$Y(s) = 2sH(s) + F(s)H(s)$$

که در آن

$$H(s) = \frac{1}{s^2 + 1}, \quad \text{و} \quad F(s) = \mathcal{L}[5e^{-t}] = \frac{5}{s+1}$$

بنابراین،

$$Y(s) = \frac{2s}{s^2 + 1} + \frac{5}{(s+1)(s^2 + 1)} \quad (25.7)$$

برای جمله دوم معادله (۲۵.۷)، لازم است تا آنرا با روش کسرهای جزئی تجزیه کنیم:

$$\begin{aligned} \frac{5}{(s+1)(s^2 + 1)} &= \frac{A}{s+1} + \frac{Bs+C}{s^2 + 1} \\ &= \frac{A(s^2 + 1) + (Bs + C)(s + 1)}{(s+1)(s^2 + 1)} \end{aligned}$$

با تطبیق جملات صورت،

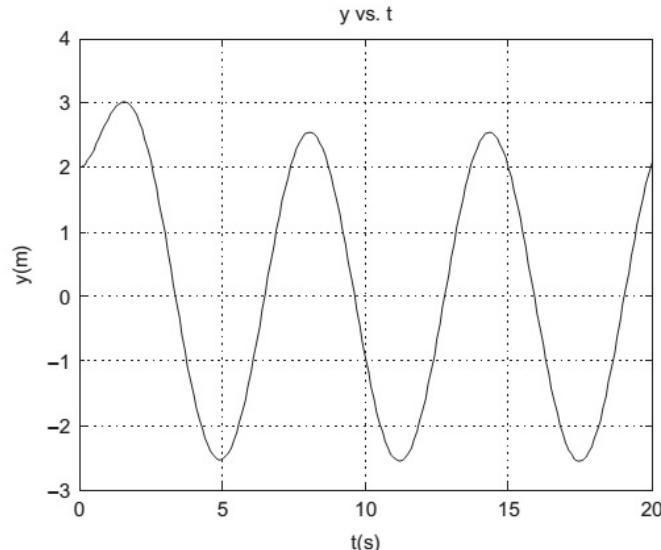
$$\begin{aligned} A + B = 0, \quad C + B = 0, \quad A + C = 5 \\ \rightarrow \quad A = \frac{5}{2}, \quad B = -\frac{5}{2}, \quad C = \frac{5}{2} \end{aligned}$$

در نتیجه،

$$\frac{5}{(s+1)(s^2 + 1)} = \frac{5}{2(s+1)} - \frac{5(s-1)}{2(s^2 + 1)}$$

و با قرار دادن در رابطه (۲۵.۷)، خواهیم داشت

$$Y(s) = \frac{2s}{s^2 + 1} + \frac{5}{2(s+1)} - \frac{5s}{2(s^2 + 1)} + \frac{5}{2(s^2 + 1)}$$



شکل ۳.۷: رسم y در مقابل t برای مثال ۳.۷

از جدول مشاهده می‌کنیم که

$$\mathcal{L}^{-1} \frac{s}{s^2 + 1} = \cos t$$

$$\mathcal{L}^{-1} \frac{1}{s^2 + 1} = \sin t$$

بنابراین، با اعمال تبدیل لاپلاس معکوس، جواب معادله (۲۴.۷) برابر است با

$$y(t) = -\frac{1}{2} \cos t + \frac{5}{2} \sin t + \frac{5}{2} e^{-t}$$

نمودار y بر حسب t در شکل (۳.۷) نشان داده شده است.

مثال ۴.۷

$$\text{جواب معادله دیفرانسیل زیر را تعیین کنید.} \\ (26.7) \quad y'' + 3y' + 2y = 5 \sin 2t, \quad y(0) = 1, \quad y'(0) = -4$$

با استفاده از همان معادله (۲۲.۷)،

$$p = 3, \quad q = 2, \quad f(t) = 5 \sin 2t = 10 \frac{\sin 2t}{2}$$

در حوزه لاپلاس،

$$H(s) = \frac{1}{s^2 + 3s + 2} = \frac{1}{(s+2)(s+1)}$$

$$F(s) = 10 \mathcal{L}\left[\frac{\sin 2t}{2}\right] = \frac{10}{s^2 + 4}$$

در نتیجه،

$$\begin{aligned} Y(s) &= [(s+3)(1)-4]H(s) + F(s)H(s) \\ &= \frac{s}{(s+2)(s+1)} - \frac{1}{(s+2)(s+1)} + \frac{10}{(s^2+4)(s+2)(s+1)} \end{aligned} \quad (27.7)$$

اکنون از تابع residue متلب، که می‌تواند با استفاده از بسط کسرهای جزئی آنرا حل کند، استفاده می‌کنیم. تابع residue دو آرگومان می‌گیرد که بصورت آرایه بوده و شامل ضرایب چندجمله‌ای‌های صورت و مخرج کسر مورد نظری است که باید بسط داده شود. سپس ضرایب صورت ("صفرها"، "ریشه‌های مخرج ("قطب‌ها")" و بخش‌هایی از غیرکسرها (که همیشه صفر است زیرا درجه صورت از مخرج کمتر است) را بما برمی‌گرداند.

برای حل کردن با residue، اولین جمله رابطه (۲۷.۷) را بصورت چندجمله‌ای بازنویسی می‌کنیم.

$$\frac{s}{(s+2)(s+1)} = \frac{s}{s^2 + 3s + 2} \quad (28.7)$$

با استفاده از کسرهای جزئی، خواهیم داشت

$$\frac{s}{(s+2)(s+1)} = \frac{2}{s+2} - \frac{1}{s+1}$$

اکنون، بیایید از residue متلب برای خلاصه کردن سمت راست رابطه (۲۸.۷) به جمع دو کسر استفاده کنیم. صورت کسر سمت راست رابطه (۲۸.۷)، مشاهده می‌شود که جمله چندجمله‌ای اول (برای s^1) دارای ضریب یک، و جمله دوم (برای s^0) صفر است. این در متلب بصورت $[0 \quad 1]$ بیان می‌شود. بهمین نحو، چندجمله‌ای مخرج کسر دارای ضریب s^2 برابر یک، برای s^1 برابر ۳ و برای s^0 برابر ۲ بوده که بصورت $[1 \quad 3 \quad 2]$ بیان می‌شود. بنابراین، متلب بصورت زیر اجرا می‌شود

```
>> [r, p, k] = residue([1 0], [1 3 2])
r =
2
-1
p =
-2
-1
k =
[]
```

این نتیج نشان میدهد که باقیماندها برابر 2 و -1 ، و قطب‌های مربوطه -2 و -1 هستند.

$$\frac{s}{s^2 + 3s + 2} = \frac{2}{s+2} - \frac{1}{s+1} \quad (29.7)$$

که همان است که با استفاده از کسرهای جزئی بدست آمد.

برای جمله دوم معادله (۲۷.۶)، ما residue([-1], [1 3 2]) را اجرا می‌کنیم و خواهیم داشت

$$\begin{aligned} \frac{-1}{(s+2)(s+1)} &= \frac{-1}{s^2 + 3s + 2} \\ &= \frac{1}{s+2} - \frac{1}{s+1} \end{aligned} \quad (30.7)$$

برای جمله سوم معادله (۲۷.۷)، ما چندجمله‌ای را بصورت زیر باز نویسی می‌کنیم:

$$\frac{10}{(s^2 + 4)(s + 2)(s + 1)} = \frac{10}{s^4 + 3s^3 + 6s^2 + 12s + 8}$$

سپس، در متلب بدست می‌آید

```
>> [r, p, k] = residue([10], [1 3 6 12 8])
r =
-0.3750 + 0.1250i
-0.3750 - 0.1250i
-1.2500
2.0000
p =
0.0000 + 2.0000i
0.0000 - 2.0000i
-2.0000
-1.0000
k =
[]
```

باقیماندها 2.0 ، -1.25 ، $-0.375 \pm 0.125j$ بوده و قطب‌های مربوطه در $j\pm 2j$ ، -2 ، -1 هستند. لذا،

$$\begin{aligned} \frac{10}{(s^2 + 4)(s + 2)(s + 1)} &= \frac{10}{s^4 + 3s^3 + 6s^2 + 12s + 8} \\ &= \frac{-0.375 + 0.125j}{s + 2j} + \frac{-0.375 - 0.125j}{s - 2j} \\ &\quad - \frac{1.25}{s + 2} + \frac{2}{s + 1} \end{aligned} \quad (31.7)$$

$$\mathcal{L}^{-1} \frac{1}{s+2} = e^{-2t}$$

$$\mathcal{L}^{-1} \frac{1}{s+1} = e^{-t}$$

$$\mathcal{L}^{-1} \frac{1}{s-2j} = e^{2jt}$$

$$\mathcal{L}^{-1} \frac{1}{s+2j} = e^{-2jt}$$

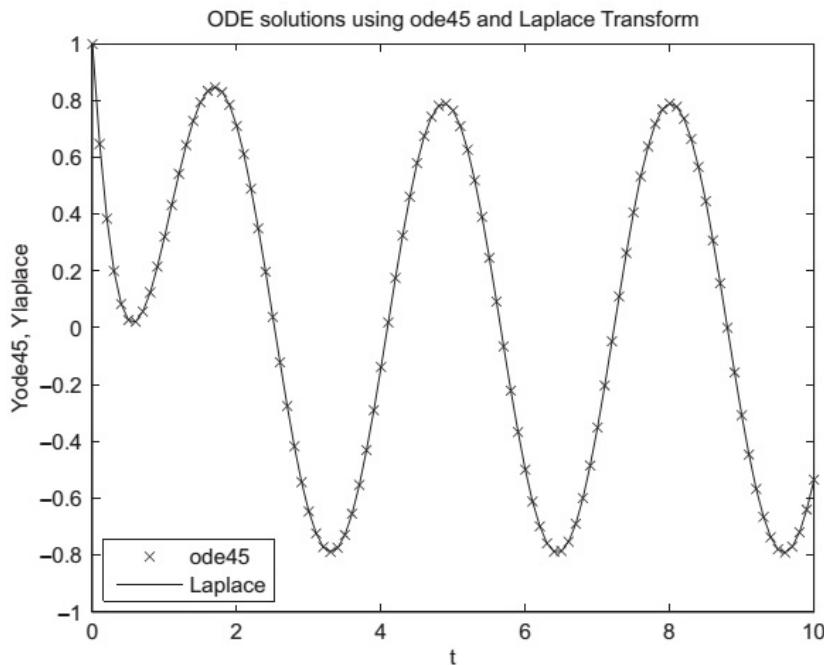
با جمع تمام جمله‌ها در $(Y(s))$ و اعمال تبدیل لاپلاس معکوس، جواب کامل رابطه (۲۶.۷) بدست می‌آید:

$$y(t) = \mathcal{L}^{-1}[Y(s)] = 1.75e^{-2t} - 0.375e^{2jt} = 0.375e^{-2jt} - 0.125je^{-2jt}$$

$$\begin{aligned} & \text{اتحادهای } e^{j\omega t} - e^{-j\omega t} = 2j \sin \omega t \quad e^{j\omega t} + e^{-j\omega t} = 2 \cos \omega t \\ & y(t) = 1.75e^{-2t} - 0.75 \cos 2t - 0.25 \sin 2t \end{aligned} \quad (۳۲.۷)$$

بهتر است برای روشن شدن مطلب، جواب بدست آمده توسط لاپلاس را با حل مستقیم معادله (۲۶.۷) توسط تابع `ode45` متلب مقایسه کنیم. برنامه زیر، و خروجی آن در شکل (۴.۷) نشان داده شده است.

```
% Example_7_4.m: This program solves Equation 7.26 by
% converting it into a system of two differential
% equations and solving with ode45.
% Let v=y', Y=[y v], and Y'=[y' v'].
% Y(1)=y, Y(2)=v, Y'(1)=Y(2), Y'(2)=5*sin 2t-3Y(2)-2Y(1)
% Initial conditions: y(0)=1.0, y'(0)=v(0)=-4.0
clear; clc;
% Define initial conditions:
Y0=[1.0 -4.0];
% Define time interval to solve:
dt=0.1; Tstop=10; Tspan=0.0:dt:Tstop;
% solve the ODE directly with ode45:
[t,Yode45]=ode45('dydt',Tspan,Y0);
% compute the solution found via Laplace Transforms
```



شکل ۴.۷: مقایسه بین جواب حاصل از تبدیل لاپلاس و تابع `ode45` مثال ۴.۷

```

%(Equation 7.32):
Ylaplace = 1.75*exp(-2*Tspan) ...
-0.75*cos(2*Tspan) -0.25*sin(2*Tspan);
plot(t,Yode45(:,1),'x',Tspan,Ylaplace),
axis([0 Tstop -1 1]);
xlabel('t') ,ylabel('Yode45,Ylaplace');
title('ODE solutions using ode45 and Laplace Transform');
legend('ode45','Laplace','Location','SouthWest');
%-----
% dydt: this function works with Example 7.4.
function Yprime=dydt(t,Y)
Yprime=zeros(2,1);
Yprime(1)=Y(2);
Yprime(2)=5*sin(2*t)-3.0*Y(2)-2.0*Y(1);

```

مثال ۵.۷

معادله دیفرانسیل زیر را حل کنید:

$$y'' + 3y' + 2y = \begin{cases} 5t, & \text{برای } t < 2 \\ 0, & \text{برای } t \geq 2 \end{cases} \quad (33.7)$$

$$y(0) = 1, \quad y'(0) = 0$$

معادله استاندارد (۲۲.۷) را بکار می‌بریم، در اینجا $q = 2$ و $p = 3$

$$f(t) = \begin{cases} 5t, & \text{برای } t < 2 \\ 0, & \text{برای } t \geq 2 \end{cases}$$

در نظر می‌گیریم. تابع $f(t)$ را بصورتی که قابلیت تبدیل لاپلاس داشته باشد، یعنی بر حسب تابع پله واحد، می‌نویسیم:

$$\begin{aligned} r(t) &= 5t[u(t) - u(t-2)] \\ &= 5tu(t) - 5tu(t-2) \end{aligned} \quad (34.7)$$

جمله اول در معادله (۳۴.۷) براحتی قابل تبدیل است، اما جمله دوم آنقدرها ساده نیست. البته، برای جمله دوم، فرض کنید $5(t-2+2)u(t-2) = 5t = 5(t-2+2)$. در این صورت

$$\begin{aligned} f(t) &= 5tu(t) - 5(t-2+2)u(t-2) \\ &= 5tu(t) - 5(t-2)u(t-2) - 5(2)u(t-2) \end{aligned}$$

در این صورت،

$$\begin{aligned} F(s) &= \mathcal{L}[5tu(t) - 5(t-2)u(t-2) - 10u(t-2)] \\ &= \frac{5}{s^2} - e^{-2s}\frac{5}{s^2} - e^{-2s}\frac{10}{s} \end{aligned}$$

بنابراین،

$$Y(s) = [(s+3)(1)H(s) + \left(\frac{5}{s^2} - e^{-2s}\frac{5}{s^2} - e^{-2s}\frac{10}{s}\right)H(s)]$$

$$H(s) = \frac{1}{s^2 + 3s + 2} = \frac{1}{(s+2)(s+1)}$$

$$Y(s) = \frac{s}{(s+2)(s+1)} + \frac{3}{(s+2)(s+1)} + \frac{5}{s^2(s+2)(s+1)} - \frac{5e^{-2s}}{s^2(s+2)(s+1)} - \frac{10e^{-2s}}{s(s+2)(s+1)}$$

فرض کنید $Y(s) = Y_1 + Y_2 + Y_3 + Y_4 + Y_5$
 $Y_1 = \frac{s}{(s+2)(s+1)}$

$$Y_2 = \frac{3}{(s+2)(s+1)}$$

$$Y_3 = \frac{5}{s^2(s+2)(s+1)}$$

$$Y_4 = -\frac{5e^{-2s}}{s^2(s+2)(s+1)}$$

$$Y_5 = -\frac{10e^{-2s}}{s(s+2)(s+1)}$$

با بسط کسرهای جزئی از طریق تابع residue مطلب، داریم

$$\begin{aligned}\mathcal{L}[Y_1] &= 2e^{-2t} - e^{-t} \\ \mathcal{L}[Y_2] &= -3e^{-2t} + 3e^{-t} \\ \mathcal{L}[Y_3] &= \frac{5}{2}t - \frac{15}{4} - \frac{5}{4}e^{-2t} + 5e^{-t} \\ \mathcal{L}[Y_4] &= -\frac{5}{2}t - t - \frac{15}{4} - \frac{5}{4}e^4e^{-2t} + 5e^2e^{-t}u(t-2) \\ \mathcal{L}[Y_5] &= -10[\frac{1}{2} + \frac{1}{2}e^4e^{-2t} - e^2e^{-t}]u(t-2)\end{aligned}$$

جمع این پنج جمله میدهد

$$y(t) = \begin{cases} = \frac{5}{2}t - \frac{15}{4} - \frac{9}{4}e^{-2t} + 7e^{-t}, & \text{برای } t < 2 \\ = -\frac{9}{4} + \frac{15}{4}e^4e^{-2t} + (7 + 5e^2)e^{-t}, & \text{برای } t \geq 2 \end{cases} \quad (35.7)$$

این مسئله را نیز می‌توان عددی با استفاده از تابع ode45 مطلب حل کرد. برنامه مطلب برای این مسئله چنین است:

```
% Example_7_5.m: This program solves Equation 7.33 by
% converting it into a system of two differential
% equations and solving with ode45.
% Let v=y', Y=[y v], and Y'=[y'v'].
% Y(1)=y, Y(2)=v, Y'(1)=Y(2).
% Y'(2)=5*t-3Y(2)-2Y(1) for t < 2
% Y'(2)=-3Y(2)-2Y(1) for t >= 2
% Initial conditions: y(0)=1,y'(0)=0
clear; clc;
% Define initial conditions:
Y0=[1.0 0.0];
% Define time interval to solve:
dt=0.1; Tstop=4; Tspan=0:dt:Tstop;
% Solve the ODE directly with ode45:
```

```
[t,Yode45]=ode45('dydt2',Tspan,Y0);
% Compute the solution found via Laplace Transforms
% (Equation 7.35)
for i=1:length(Tspan)
if Tspan(i) <2
Ylaplace(i)=2.5*Tspan(i)-15/4 -...
9/4*exp(-2*Tspan(i))+7*exp(-Tspan(i));
else Ylaplace(i)= -(9/4+15/4*exp(4)) ...
* exp(-2*Tspan(i)) ...
+(7+5*exp(2))*exp(-Tspan(i));
end
end
plot(t,Yode45(:,1),'x',Tspan,Ylaplace),
title('ODE solutions using ode45 and Laplace Transform');
legend('ode45','Laplace');

-----
% dydt2: This function works with Example 7.5.
% Y(1)=y,Y(2)=v
% Y1'=Y(2),Y2'=5*t-3Y2-2Y1,if t >2,Y2'=-3Y2-2Y1,if t >= 2
function Yprime=dydt2(t,Y)
Yprime=zeros(2,1);
Yprime(1)=Y(2);
if t <2
Yprime(2)=5*t-3.0*Y(2)-2.0*Y(1);
else
Yprime(2)=-3.0*Y(2)-2.0*Y(1);
end
```

مقایسه ode45 با تبدیل لاپلاس در شکل (۵.۷) نشان داده شده است.

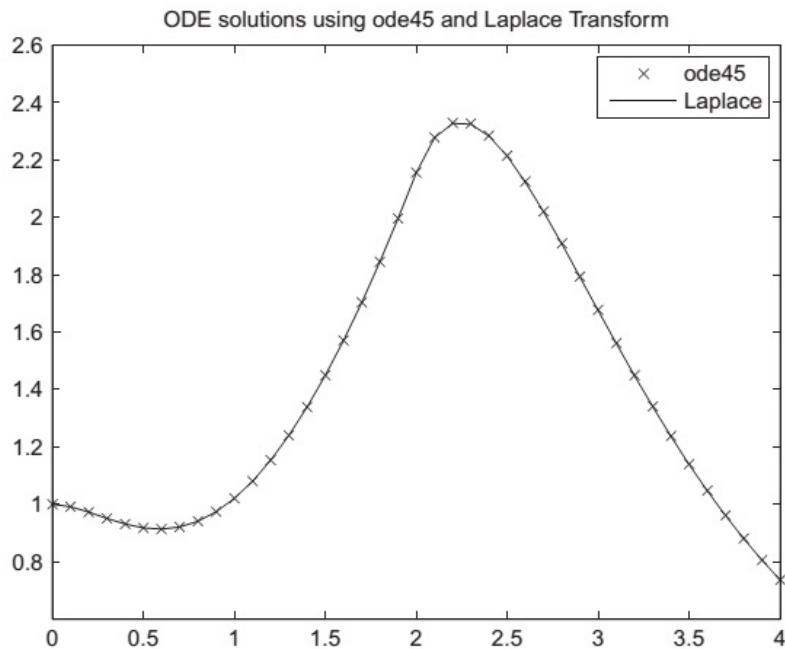
۵.۷ کانولوشن

دوتابع $f(t)$ و $g(t)$ بترتیب با تبدیل لاپلاس $F(s)$ و $G(s)$ داده شده است. در این صورت تبدیل لاپلاس معکوس حاصل ضرب $F(s)G(s)$ برابر انتگرال کانولوشن خواهد بود

$$\mathcal{L}^{-1}[F(s)G(s)] = \int_0^t f(\tau)g(t-\tau)d\tau = \int_0^t g(\tau)f(t-\tau)d\tau \quad (36.7)$$

اثبات:

$$\begin{aligned} F(s)G(s) &= \left[\int_0^\infty f(p)e^{-ps}dp \right] \left[\int_0^\infty g(\tau)e^{-\tau s}d\tau \right] \\ &= \int_0^\infty g(\tau) \left[\int_0^\infty f(p)e^{-(p+\tau)s}dp \right] d\tau \end{aligned} \quad (37.7)$$

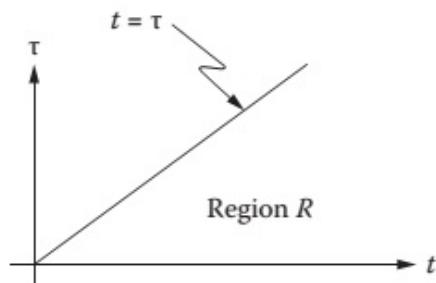


شکل ۵.۷: مقایسه جواب تبدیل لاپلاس با جواب تابع ode45 متبہ برای مثال ۵.۶

فرض کنید $t = p + \tau$ باشد. در این صورت، وقتی $p = 0$ و وقتی $t = \tau$ و وقتی $p = \infty$ در نتیجه خواهد بود. ضمناً، $dp = dt$ است. بنابراین،

$$\begin{aligned} F(s)G(s) &= \int_0^\infty g(\tau) \left[\int_0^\infty f(t - \tau) e^{-ts} dt d\tau \right] \\ &= \iint_R g(\tau) f(t - \tau) e^{-ts} dt d\tau \end{aligned} \quad (۳۸.۷)$$

که در آن R ناحیه زیر خط $t = \tau$ است که در شکل (۴.۷) نشان داده شده است. مرتبه انتگرال‌گیری انتگرال چندگانه، بطوری که در معادله (۳۸.۷) نوشته شده است، ابتدا نسبت به t ، سپس نسبت به τ انتگرال‌گیری می‌شود. در این حالت، مراتب انتگرال‌گیری زیاد اهمیتی ندارد، بنابراین میتوانیم ابتدا نسبت به τ و سپس نسبت به t انتگرال‌گیری نمائیم. این میدهد



شکل ۷.۶: ناحیه انتگرال‌گیری برای کانولوشن در صفحه (τ, t) .

$$\begin{aligned} F(s)G(s) &= \iint_R g(\tau)f(t-\tau)e^{-ts}dt d\tau \\ &= \int_0^\infty e^{-ts} \int_0^t g(\tau)f(t-\tau)d\tau dt \\ &= \mathcal{L} \left[\int_0^t g(\tau)f(t-\tau)d\tau \right] \end{aligned}$$

بنابراین،

$$\mathcal{L}^{-1}[F(s)G(s)] = \int_0^t g(\tau)f(t-\tau)d\tau \quad (۳۹.۷)$$

برعکس، می‌توان جای f و g را عوض کرد. خواهد شد

$$\mathcal{L}^{-1}[F(s)G(s)] = \int_0^t g(t-\tau)f(\tau)d\tau$$

کانونلوشن را با عملگر “*” نشان میدهند؛ لذا،

$$\mathcal{L}^{-1}[F(s)G(s)] = f(t) * g(t) = g(t) * f(t)$$

مثال ۴.۷

تبدیل لاپلاس را برای تابع کانونلوشن با مطالعه شکل (۴.۵)(الف) که در آن دو پالس $f(t)$ و $g(t)$ را داریم نشان خواهیم داد:

$$f(t) = \begin{cases} \alpha, & 0 \leq t \leq T_1 \\ 0 & \text{در غیر این صورت} \end{cases} \quad \text{برای}$$

$$g(t) = \begin{cases} \beta, & T_2 \leq t < T_3 \\ 0 & \text{در غیر این صورت} \end{cases} \quad \text{برای}$$

شکل (۴.۵)(ب) عملکرد کانونلوشن که در آن تصویر آئینه‌ای $f(\tau) - f(t)$ تابع $f(\tau)$ و در طول محور t می‌لغزد (با تغییر τ) در حالی که انتگرال گیری دو تابع انجام می‌شود، نشان می‌دهد. جواب کانونلوشن یک شکل موج ذوزنقه‌ای است که در شکل (۴.۵)(ج) رسم شده است.

$$f(t) * g(t) = \begin{cases} \frac{\alpha\beta}{(T_3 - T_2)}(t - T_2), & T_2 \leq t \leq T_3 \\ \frac{\alpha\beta}{(T_3 - T_2)}, & T_3 \leq t \leq (T_1 + T_2) \\ \frac{-\alpha\beta}{(T_3 - T_2)}(t - T_1 - T_3) & (T_1 + T_2) \leq t \leq (T_1 + T_3) \\ 0 & \text{در غیر این صورت} \end{cases} \quad (۴۰.۷)$$

اکنون تبدیل لاپلاس $f(t)$ و $g(t)$ را محاسبه کرده، آنها را در حوزه s ضرب کرده، و سپس تبدیل لاپلاس معکوس می‌گیریم. جواب با قبل یکسان خواهد بود. در قبل تبدیل لاپلاس یک پالس را در معادله (۱۷.۷) محاسبه کردیم. بنابراین

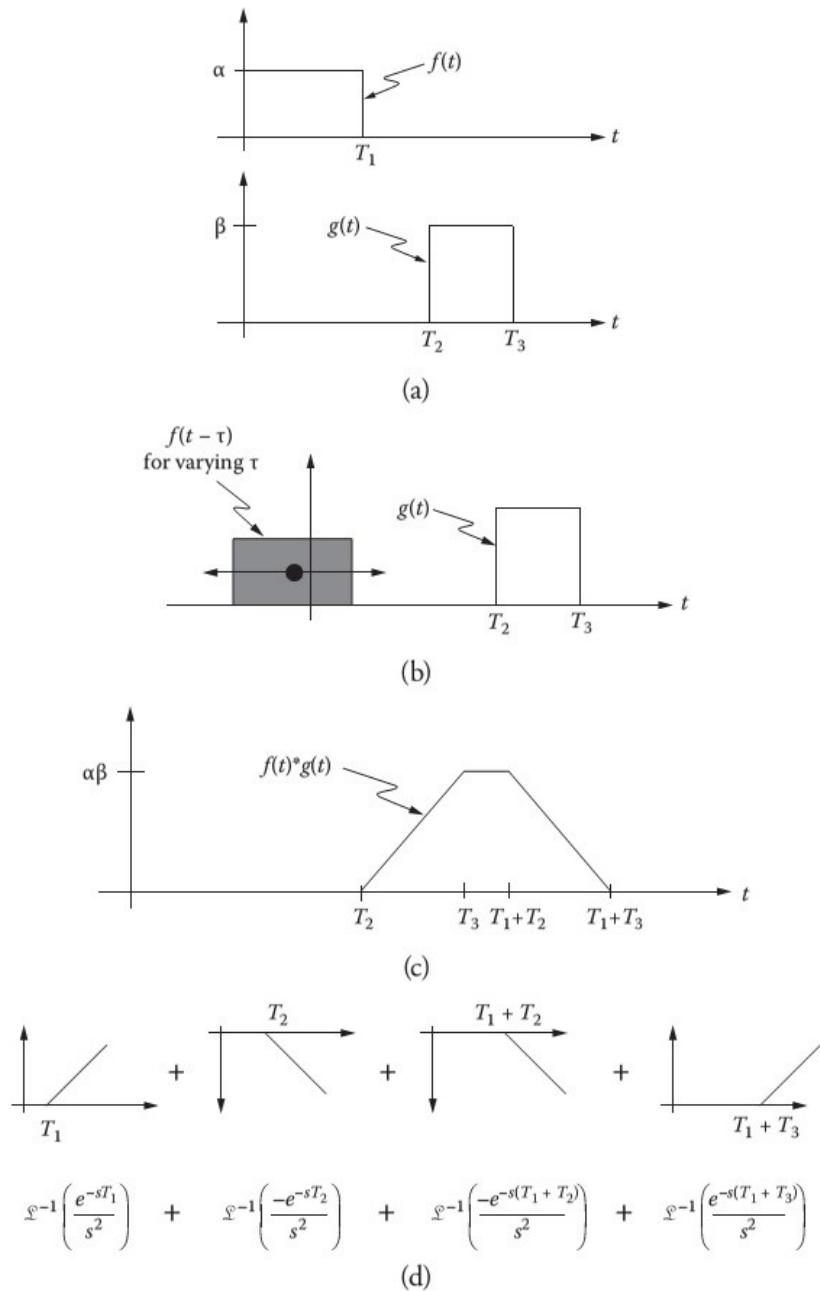
$$F(s) = \frac{\alpha}{s}(1 - e^{-sT_1})$$

$$G(s) = \frac{\beta}{s}(e^{-sT_2} - e^{-sT_3})$$

و ضرب آنها

$$F(s)G(s) = \frac{\alpha\beta}{s^2} \left[e^{-sT_2} - e^{-sT_3} - e^{-s(T_1+T_2)} + e^{-s(T_1+T_3)} \right] \quad (۴۱.۷)$$

با کمی تفحص، تبدیل معکوس رابطه (۴۱.۷) شامل جمع چهار تابع شبیدار با تاخیر زمانی است که بصورت تصویری در شکل (۴.۵)(د) نشان داده شده است. جمع این چهار تابع همان شکل موج ذوزنقه‌ای (۴.۵)(ج) خواهد بود.



شکل ۷.۷: نمایش تصویری کانولوشن. (الف) شکل موج پالسهای $f(t)$ و $g(t)$. (ب) با تصویر سازی آئینه‌ای $f(t - \tau)$ و لغزیدن در طول محور t و انتگرال‌گیری با $\int g(\tau) f(t - \tau) d\tau$. (ج) جواب تصویری کانولوشن $f(t) * g(t)$ یک شکل موج ذوزنقه‌ای است. (د) همین جواب را می‌توان با گرفتن تبدیل لاپلاس معکوس $F(s)G(s)$ بدست آورد.

۶.۷ کاربرد تبدیل لاپلاس در مدارها

در نظریه مدارها، خازن‌ها و سیم‌پیچ‌ها روابط سازنده‌ای دارند که باعث می‌شود بتوان آنها را در حوزه لاپلاس تحلیل نمود. با خازن شروع می‌کنیم،

$$i_C = C \frac{dv_C}{dt}$$

در اینجا i_C جریان خازن، v_C ولتاژ خازن، و C مقدار ظرفیت (بر حسب فاراد) خازن هستند. با اعمال رابطه (۱۸.۷)، می‌توانیم این معادله را در حوزه لاپلاس بنویسیم:

$$I_C(s) = C[sV_C(s) - v_C(0)]$$

اگر در این لحظه از شرایط اولیه چشم‌پوشی کنیم، خواهیم داشت

$$V_C(s) = I_C(s) \frac{1}{Cs} \quad (۴۲.۷)$$

توجه کنید که رابطه (۴۲.۷) به قانون اهم ($v = iR$) شبیه است، و از این‌رو امپدانس مختلط Z_C را برای خازن تعریف می‌کنیم

$$Z_C(s) = \frac{1}{Cs} \quad (۴۳.۷)$$

با تحلیل مشابه رابطه سازنده سیم‌پیچ، $v_L = L \frac{di_L}{dt}$ ، امپدانس مختلط Z_L برای خودالقائی L (با صرفنظر کردن از شرایط اولیه) می‌توان نشان داد که

$$Z_L(s) = Ls \quad (۴۴.۷)$$

سرانجام برای مقاومت، تبدیل لاپلاس قانون اهم خواهد بود

$$Z_R(s) = R \quad (۴۵.۷)$$

چون قبلًا ثابت کردیم تبدیل لاپلاس خطی است، می‌توانیم مدارهای خطی مشکل را مستقیماً در حوزه لاپلاس با استفاده از همان روش جبری که برای روابط ورودی و خروجی در مدارات فقط مقاومتی بکار برده می‌شود، حل کنیم. برای مثال، مدار RLC موازی شکل (۸.۷) را با جریان ورودی ($I_0(s)$ ، ولتاژ خروجی ($V_C(s)$ ، و امپدانس‌های مختلط در نظر گیرید. خواهیم داشت:

$$V_C(s) = I_0(s)[Z_R(s)||Z_C(s)||Z_L(s)]$$

در اینجا عملگر (||) نمایشگر محاسبه امپدانس بصورت موازی است. در این صورت،

$$\begin{aligned} \frac{V_C(s)}{I_0(s)} &= \frac{1}{\frac{1}{Z_R} + \frac{1}{Z_C} + \frac{1}{Z_L}} \\ &= \frac{1}{\frac{1}{R} + Cs + \frac{1}{Ls}} \\ &= \frac{1}{C} \times \frac{s}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \end{aligned} \quad (۴۶.۷)$$

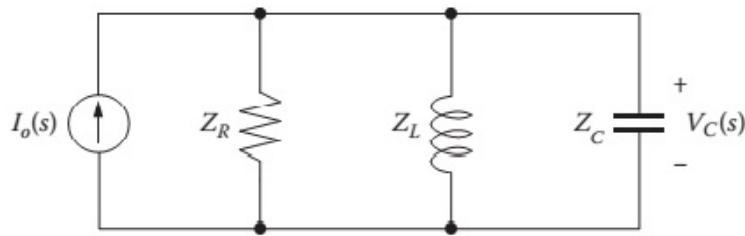
می‌توانیم تعریف کنیم

$$H(s) = \frac{V_C(s)}{I_0(s)} \quad (۴۷.۷)$$

که این را تابع تبدیل مدار، یعنی، نسبت متغیر خروجی (در اینجا V_C) به تابع ورودی (در اینجا I_0) تعریف می‌کنیم.

اکنون بیایید فرم سینوزوئیدال برای $i_o(t)$ بصورت زیر در نظر گیریم

$$i_o(t) = Ce^{j\omega t}$$



شکل ۸.۷: مدار RLC با استفاده از امپدانس‌ها.

که در آن $e^{j\omega t} = \cos \omega t + j \sin \omega t$ جمع جمله‌های سینوسی و کسینوسی با فرکانس ω است. برای راحتی، یک ضریب ثابت C برای سادگی ریاضی به آن نیز اضافه کرده‌ایم. در این صورت،

$$I_o(s) = C \frac{1}{s - j\omega}$$

و

$$\begin{aligned} V_C(s) &= H(s)I_o(s) \\ &= \frac{s}{(s + s_1)(s + s_2)(s - j\omega)} \end{aligned} \quad (48.7)$$

که در آن s_1 و s_2 صفرهای مخرج معادله (۴۶.۷) هستند و از R , L ، و C محاسبه می‌شوند. ضمناً برای این تحلیل، فرض می‌کنیم که s_1 و s_2 حقیقی هستند. با مرتب کردن معادله (۴۸.۷) توسط کسرهای جزئی، خواهیم داشت

$$V_C(s) = \frac{\frac{s_1}{(s_1 + s_2)(s_1 - j\omega)}}{s + s_1} + \frac{\frac{s_2}{(s_1 + s_2)(s_2 - j\omega)}}{s + s_2} + \frac{\frac{j\omega}{(j\omega + s_1)(j\omega + s_2)}}{s - j\omega}$$

و با تبدیل لاپلاس معکوس، بدست می‌آوریم

$$v_C(t) = \frac{s_1}{(s_1 + s_2)(s_1 - j\omega)} e^{-s_1 t} + \frac{s_2}{(s_1 + s_2)(s_2 - j\omega)} e^{-s_2 t} + H(j\omega) e^{j\omega t} \quad (49.7)$$

همچنین معادله (۴۹.۷) کمی غامض و مشکل بنظر میرسد، توجه کنید که اولین دو جمله شامل نمائی منفی و در نتیجه در حالت ماندگار^۴ ($t \rightarrow \infty$)، این معادله بصورت زیر تقلیل می‌یابد:

$$\lim_{t \rightarrow \infty} v_C(t) = H(j\omega) e^{j\omega t} \quad (50.7)$$

مفهوم معادله (۵۰.۷) این است که برای هر ورودی سینوزوئیدالی (به فرکانس ω) به مدار، می‌توانیم مقدار خروجی حالت ماندگار (به فرکانس ω) را بوسیله $H(j\omega)$ محاسبه کنیم. این نحوه بدست آوردن را می‌توان برای تعیین پاسخ فرکانسی^۵ هر مدار خطی با پیدا کردنتابع انتقال ($H(s)$ و سپس قرار دادن $\omega = j\omega$ تعیین کرد).

پاسخ فرکانسی معمولاً با رسم دو نمودار که نشانده مقدار $|H(j\omega)|$ و فاز $\angle H(j\omega)$ هر دو بر حسب فرکانس، مشخص می‌شود.

نمودار مقداری معمولاً در مقیاس محور لگاریتمی-لگاریتمی و نمودار فاز در مقیاس نیمه لگاریتمی (خطی در مقابل فرکانس لگاریتمی) رسم می‌شود. هردو اینها را نمودار بود^۶ می‌نامند.

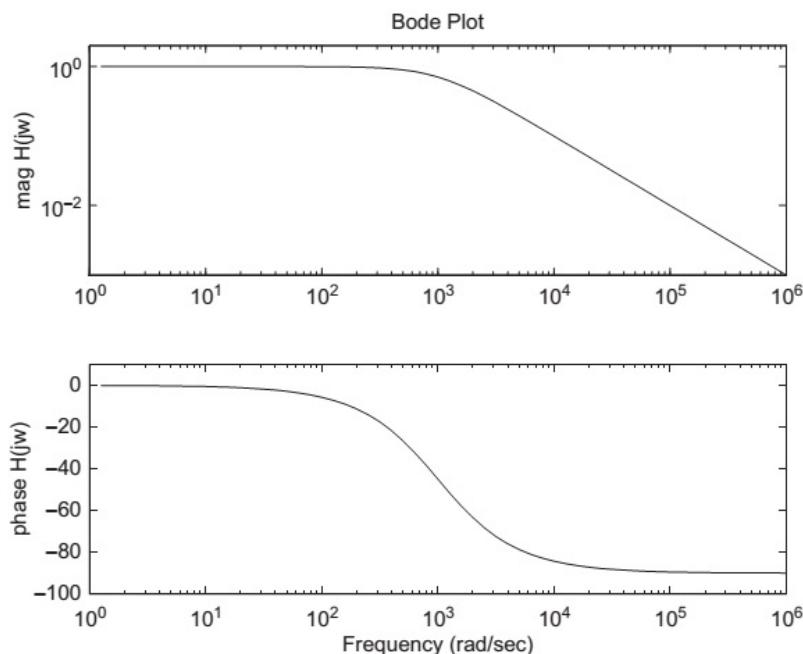
^۴steady state^۵frequency response^۶Bode plot

مثال ۷.۷

فرض کنید

$$H(s) = \frac{1000}{s + 1000}$$

نمودار مقداری و فازی $H(j\omega)$ را در بازه $\omega = [1 \times 10^{-1}, 1 \times 10^6]$ رادیان بر ثانیه رسم کنید. جواب آن در شکل (۹.۷) ترسیم شده است.



شکل ۹.۷: نمودار بود برای $H(j\omega) = \frac{1000}{j\omega + 1000}$

```
% Example_7_7.m: Plot magnitude and phase of
% H(jw)= 1000/(s+1000) for w=[1e-1,1e6].
% First,create a logarithmic point set for the frequency
% range from 1 to 1e6 rad/sec with ten points in each
% decade.
for k=1:60
w(k)=10^(k/10);
end
% Compute the magnitude for H(jw)
H_mag = abs( 1000 ./ (j*w + 1000) );
% Compute the phase for H(jw),and convert to degrees
H_phase = (180/pi) * angle( 1000 ./ (j*w + 1000) );
% Plot magnitude on log-log axes
subplot(2,1,1);
loglog(w,H_mag);
```

```

axis([10^0 10^6 10^-3 2*10^0]);
ylabel('mag H(jw)');
title('Bode Plot');
% Plot phase on log-linear axes
subplot(2,1,2);
semilogx(w,H_phase);
axis([10^0 10^6 -100 10]);
xlabel('Frequency (rad/sec)');
ylabel('phase H(jw)');

```

۷.۷ پاسخ ضربه

در بخش ۷.۲.۷، تابع دلتا و خواصش را بحث کردیم. اگر تابع دلتا را به ورودی یک مداری که تابع انتقال $H(s)$ آن معلوم است اعمال کنیم، چه اتفاق می‌افتد؟ در معادله (۴۷.۷)، ما $H(s)$ را بصورت نسبت خروجی مدار به ورودی مدار تعریف کردیم. اگر فرض کنیم ورودی تابع دلتا باشد، در این صورت

$$\begin{aligned}
V_C(s) &= H(s)I_o(s) \\
&= H(s)(1)
\end{aligned} \tag{۵۱.۷}$$

که در آن $i_o(t) = \delta(t)$ فرض کردیم و تبدیل لاپلاس آن واحد است. اگر از رابطه (۵۱.۷) تبدیل معکوس بگیریم، خواهیم داشت

$$v_C(t) = h(t)$$

بعارت دیگر، اگر ورودی مدار تابع دلتا باشد، در آن صورت خروجی مدار تبدیل معکوس تابع انتقال $h(t)$ خواهد بود و آن به پاسخ ضربه‌ای^۷ معروف است. بر عکس، اگر دقیقاً پاسخ ضربه‌ای $h(t)$ مدار را محاسبه (و یا اندازه‌گیری) کنیم، در آن صورت می‌توانیم تابع انتقال $H(s)$ را بدست آورده و رفتار مدار را برای هر شکل موج دلخواهی تعیین و پیشگوئی نمائیم. این مفهوم در مثال ۸.۷ نشان داده شده است.

مثال ۸.۷:

مداری دارای پاسخ ضربه‌ای زیر است

$$h(t) = 10e^{-2t} \cos 10t$$

پاسخ پله‌واحد مدار $r(t)$ چیست؟
با تبدیل $h(t)$ ، تابع انتقال $H(s)$ بدست می‌آید.

$$H(s) = 10 \frac{s+2}{(s+2)^2 + 100}$$

که در آن معادلات (۹.۷) و (۱۱.۷) را برای تبدیل لاپلاس اعمال می‌کنیم. برای تعیین پاسخ پله‌واحد،

$$R(s) = 10 \frac{s+2}{(s+2)^2 + 100} \times \frac{1}{s}$$

^۷impulse response

اکنون تبدیل معکوس را بصورت زیر محاسبه می‌کنیم:

$$\begin{aligned}
 r(t) &= \mathcal{L}^{-1}\{R(s)\} \\
 &= 10\mathcal{L}^{-1}\left[\frac{s+2}{s^3+4s^2+104s}\right] \\
 &= 10\mathcal{L}^{-1}\left[\frac{-0.0096-0.0481j}{s+(2-10j)} + \frac{-0.0096-0.0481j}{s+(2+10j)} + \frac{0.0192}{s}\right] \\
 r(t) &= 10[-0.0096e^{-2t+10jt} + 0.0481je^{-2t+10jt} - 0.0096e^{-2t-10jt} \\
 &\quad - 0.481je^{-2t-10jt} + 0.0192u(t)] \\
 &= 1.92e^{-2t}[-e^{-2t}\cos 10t - 5e^{-2t}\sin 10t + u(t)]
 \end{aligned}$$

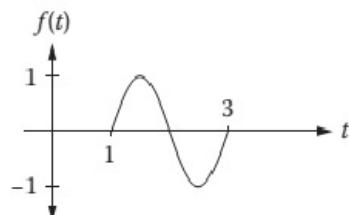
مسائل

۱.۷ • $\mathcal{L}[\sin^2 \omega t]$ مطلوبست

۲.۷ • $\mathcal{L}[e^{2t} \cos t]$ مطلوبست

۳.۷ • $\mathcal{L}[e^{-3t} \sin 2t]$ مطلوبست

۴.۷ • تبدیل لاپلاس تابعی که در شکل نشان داده شده است بدست آورید.



شکل ۱۰.۷: تابع $f(t)$ شامل یک سیکل از یک موج سینوسی در بازه $t \leq 3$ ثانیه است.

۵.۷ • $\mathcal{L}^{-1}\left[\frac{2s+1}{(s^2+9)(s-2)}\right]$ مطلوبست

۶.۷ • $\mathcal{L}^{-1}\left[\frac{1}{(s^2+5s+6)}\right]$ مطلوبست

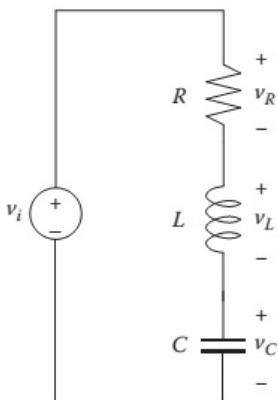
۷.۷ • $\mathcal{L}^{-1}\left[\frac{1}{(s+1)(s^2+3s+2)}\right]$ با استفاده از کانولوشن در حوزه زمان مطلوبست

۸.۷ • با استفاده از تبدیل لاپلاس معادله دیفرانسیل زیر را حل کنید.
 $y'' + 3y' - 2y = 3e^{-t}, \quad y(0) = 1, \quad y'(0) = 2$

پروژه‌ها

۱.۷ پروژه

شکل (۱۱.۴) یک مدار سری RLC را که با ولتاژ ورودی $v_i(t)$ تغذیه و سه ولتاژ خروجی $v_R(t)$ ، $v_L(t)$ و $v_C(t)$ برای آن تعریف می‌شود، نشان میدهد.



شکل ۱۱.۷: مدار RLC سری.

۱. با استفاده از امپدانس مختلط، سه تابع انتقال $H_R(s) = \frac{V_L(s)}{V_i(s)}$ ، $H_R(s) = \frac{V_R(s)}{V_i(s)}$ ، و

$H_R(s) = \frac{V_C(s)}{V_i(s)}$ را بدست آورید. تابع انتقال‌ها را بصورت استاندارد زیر بنویسید:

$$H(s) = \frac{K_1 s^2 + K_2 s + K_3}{s^2 + K_4 s + K_5}$$

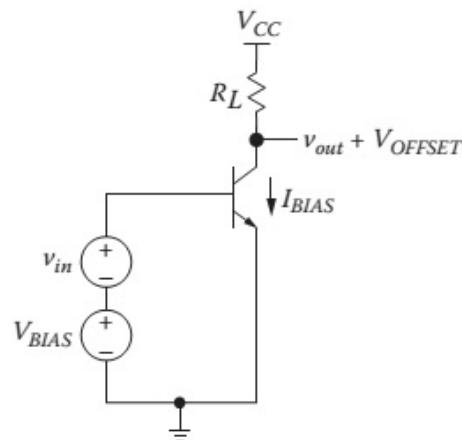
که در آن K_1 تا K_5 ثابت‌های حقیقی هستند. توجه کنید که این پنج ثابت به نحوی مقیاس بندی می‌شوند که بزرگترین مرتبه جمله در مخرج مقدار ثابتی ندارد.

۲. نمودارهای بودی را برای $(H_L(j\omega), H_R(j\omega), H_C(j\omega))$ رسم کنید. مقادیر عناصر را $C = 0.1 \mu F$ ، $L = 0.1 mH$ ، و $R = 1k\Omega$ در نظر گیرید. نمودارها را در بازه فرکانسی $\omega = [1, 1 \times 10^{10}] rad/sec$ رسم نمائید.

۳. این سه تابع انتقال نمایشکر سه نوع فیلتر هستند: بالاگذر، پائین‌گذر، و میان‌گذر. هر کدام از این تابع انتقال‌ها به کدام فیلتر مربوط می‌شوند؟

۲.۷ پروژه

یک تقویت‌کننده امیتر مشترک در شکل (۱۲.۷) با ولتاژ ورودی v_{in} و ولتاژ خروجی v_{out} نشان داده شده است. در این نوع مدار تک ترانزیستور، ولتاژهای مطلوب روی مولفه DC ، V_{BIAS} ، و V_{OFFSET} سوار هستند بمنظور بیاس ترانزیستور در ناحیه خطی (یعنی تقویت) بکار می‌روند. شکل (۱۳.۷) مدل هایپرید-پی سیگنال کوچک ترانزیستور دوقطبی را نشان میدهد [۴]. در مدل ترانزیستور سیگنال کوچک، مولفه‌های DC ، V_{BIAS} ، V_{CC} ، و V_{OFFSET} چشم‌پوشی می‌شود، و می‌توانیم قانون جریان کیرشهف را برای حل تابع تبدیل بر حسب امپدانس مختلط بکار ببریم. با نوشتن معادلات KCL در گره‌های v_π و v_{out} خواهیم داشت:



شکل ۱۲.۷: مدل سیکنال کوچک تقویت کننده امیتر مشترک. مقاومت r_b مقاومت بیس سری، مقاومت r_π مقاومت بیس-امیتر سیکنال کوچک، g_m هدایت انتقالی ("بهره" ترانزیستور)، r_o مقاومت امیتر-کلکتور، و c_π و c_μ بترتیب خازنهای و بیس-امیتر کلکتور-امیتر هستند. R_L مقاومت بار تقویت کننده است.

$$\frac{(v_{in} - v_\pi)}{r_b} - \frac{v_\pi}{r_\pi \parallel \frac{1}{c_\pi s}} + \frac{(v_{out} - v_\pi)}{\frac{1}{cs}} = 0 \quad (۵۲.۷)$$

$$\frac{(v_{out} - v_\pi)}{\frac{1}{cs}} + g_m v_\pi + \frac{v_{out}}{r_o \parallel R_L} = 0 \quad (۵۳.۷)$$

که در آن v_π ولتاژ امیتر-بیس سیکنال کوچک، r_b مقاومت امیتر-بیس سیکنال کوچک، r_π مقاومت بیس؛ c_π و c_μ بترتیب ظرفیت خازنهای امیتر-بیس و کلکتور-امیتر؛ r_o مقاومت امیتر-کلکتور سیکنال کوچک؛ g_m هدایت انتقالی سیکنال کوچک؛ و R_L مقاومت بار تقویت کننده هستند. از حل این معادلاتتابع انتقال سیستم بدست می‌آید:

$$H(s) = \frac{v_{out}}{v_{in}} = \frac{\frac{1}{r_b}(cs - g_m)}{c_\pi cs^2 + \frac{c_\pi}{r_o \parallel R_L} + \frac{c}{r_o \parallel R_L \parallel r_b \parallel \frac{1}{g_m}} s + \frac{1}{(r_b \parallel r_\pi)(r_o \parallel R_L)}} \quad (۵۴.۷)$$

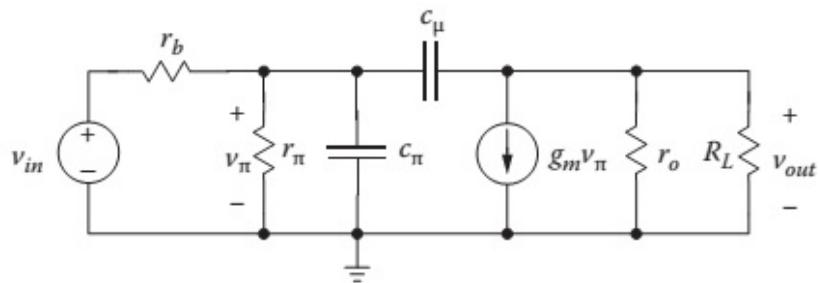
اگر فرض کنیم که ترانزیستور مدل 2N3904 است، می‌توانیم مقادیر عددی را برای پارامترهای مدار از دفترچه داده‌های سازنده برای این قطعه بدست آوریم [۶]. اینها در جدول (۱.۷) خلاصه شده است. ضمناً فرض می‌کنیم که برای بدست آوردن جریان کلکتور (I_{BIAS}) برای یک میلی آمپر و مقاومت بار $R_L = 5 k\Omega$ ، ولتاژ V_{BIAS} تنظیم شده است. چون ترانزیستور 2N3904 یک عنصر مجزاء است (جزء مدارات مجتمع نیست)، همچنین فرض میکنیم که مقاومت بیس کوچک $= 25\Omega$ و نیز $r_b = \frac{I_o}{V_T} = \frac{kT}{q} = 0.026mV$ است.

۱. نمودار بود را با استفاده از پارامترهای جدول (۱.۷) برای تابع انتقال رابطه (۵۲.۷) رسم کنید. بازه فرکانس را $[1 \times 10^5, 1 \times 10^{11}] rad/sec$ در نظر گیرید.

۲. یکی از تحلیل‌های سرانگشتی این مدار استفاده از تقریب میلر^۱ است. مقادیر دیگری برای c_μ و c_π بقرار زیر فرض می‌کنیم

$$c = 0$$

^۱Miller approximation



شکل ۱۳.۷: تقویت کننده امپیتر مشترک. ولتاژ V_{BIAS} یک ولتاژ مولفه ("DC") ورودی است، که برای تعیین نقطه کار (I_{BIAS}) مناسب ترانزیستور تغییر داده می‌شود. ولتاژ V_{OFFSET} یک مولفه DC خروجی است. ولتاژهای مطلوب مولفه‌های متغیر ورودی v_{in} و خروجی v_{out} می‌باشند.

جدول ۱.۷: پارامترهای ترانزیستور NPN بشماره 2N3904

| پارامترهای ترانزیستور | مقادیر نمونه |
|-----------------------|--------------|
| $h_{FE} (= \beta)$ | 125 |
| $C_{obo} (= c_\mu)$ | 3 pF |
| $C_{ibo} (= c_\pi)$ | 4 pF |
| $h_{ie} (= r_\pi)$ | 4 kΩ |
| $h_{oe} (= 1/r_o)$ | 10 μmho |

$$c_\pi = c_\pi + [g_m(r_o||R_L) + 1]c$$

این مقادیر تقریبی را برای c_μ و c_π در رابطه (۱۴.۷) قرار دهید و تابع انتقال تقریبی $H_{Miller}(s)$ را بدست آورید.

۳. تابع انتقال تقریبی $H_{Miller}(s)$ را روی همان محوریکه اکنون برای $H(s)$ دارید رسم کنید.
تحت چه شرایط تقریب میلر برای استفاده قابل قبول است؟

| | $f(t)$ | $F(s)$ |
|----|--|---|
| 16 | $\frac{1}{(a-b)}(e^{at} - e^{bt})$ | $\frac{1}{(s-a)(s-b)} \quad (a \neq b)$ |
| 17 | $\frac{1}{(a-b)}(ae^{at} - be^{bt})$ | $\frac{s}{(s-a)(s-b)} \quad (a \neq b)$ |
| 18 | $-\frac{(b-a)e^{at} + (c-a)e^{bt} + (a-b)e^{ct}}{(a-b)(b-c)(c-a)}$ | $\frac{1}{(s-a)(s-b)(s-c)}$ |
| 19 | $\frac{1}{a}\sin at$ | $\frac{1}{s^2 + a^2}$ |
| 20 | $\cos at$ | $\frac{s}{s^2 + a^2}$ |
| 21 | $\frac{1}{a}\sinh at$ | $\frac{1}{s^2 - a^2}$ |
| 22 | $\cosh at$ | $\frac{s}{s^2 - a^2}$ |
| 23 | $\frac{1}{a^2}(1 - \cos at)$ | $\frac{1}{s(s^2 + a^2)}$ |
| 24 | $\frac{1}{a^3}(at - \sin at)$ | $\frac{1}{s^2(s^2 + a^2)}$ |
| 25 | $\frac{1}{2a^3}(\sin at - at \cos at)$ | $\frac{1}{(s^2 + a^2)^2}$ |
| 26 | $\frac{t}{2a}\sin at$ | $\frac{s}{(s^2 + a^2)^2}$ |
| 27 | $\frac{1}{2a}(\sin at + at \cos at)$ | $\frac{s^2}{(s^2 + a^2)^2}$ |
| 28 | $t\cos at$ | $\frac{s^2 - a^2}{(s^2 + a^2)^2}$ |

جدول ٢.٧: تبدیل لاپلاس

| | $f(t)$ | $F(s)$ |
|----|--|--|
| 1 | Unit step: 1 | $\frac{1}{s}$ |
| 2 | Ramp: t | $\frac{1}{s^2}$ |
| 3 | Power of t : $\frac{t^{n-1}}{(n-1)!}$ | $\frac{1}{s^n} \quad (n = 1, 2, \dots)$ |
| 4 | Derivative: $\frac{df(t)}{dt}$ | $sF(s) - f(0)$ |
| 5 | Time delay: $f(t-T)$ | $e^{-sT} F(s)$ |
| 6 | Exponential: $e^{at} f(t)$ | $F(s-a)$ |
| 7 | Convolution: $f(t)*g(t)$ | $F(s) G(s)$ |
| 8 | Delta function: $\delta(t)$ | 1 |
| 9 | $\frac{1}{\sqrt{\pi t}}$ | $\frac{1}{\sqrt{s}}$ |
| 10 | $2\sqrt{\frac{t}{\pi}}$ | $s^{-\frac{3}{2}}$ |
| 11 | $\frac{2^n t^{n-1/2}}{1 \times 3 \times 5 \times \dots \times (2n-1)\sqrt{\pi}}$ | $s^{-\frac{n+1}{2}} \quad (n = 1, 2, \dots)$ |
| 12 | e^{at} | $\frac{1}{s-a}$ |
| 13 | $t e^{at}$ | $\frac{1}{(s-a)^2}$ |
| 14 | $\frac{1}{(n-1)!} t^{n-1} e^{at}$ | $\frac{1}{(s-a)^n} \quad (n = 1, 2, \dots)$ |
| 15 | $t^{k-1} e^{at}$ | $\frac{\Gamma(k)}{(s-a)^k} \quad (k > 0)$ |

| | $f(t)$ | $F(s)$ |
|----|--|---|
| 29 | $\frac{\cos at - \cos bt}{b^2 - a^2}$ | $\frac{s}{(s^2 + a^2)(s^2 + b^2)} \quad (a^2 \neq b^2)$ |
| 30 | $\frac{1}{b} e^{at} \sin bt$ | $\frac{1}{(s - a)^2 + b^2}$ |
| 31 | $e^{at} \cos bt$ | $\frac{s - a}{(s - a)^2 + b^2}$ |
| 32 | $e^{-at} - e^{\frac{-at}{2}} \cos \frac{at\sqrt{3}}{2} - \sqrt{3} \sin \frac{at\sqrt{3}}{2}$ | $\frac{3a^2}{s^3 + a^3}$ |
| 33 | $\sin at \cosh at - \cos at \sinh at$ | $\frac{4a^3}{s^4 + 4a^4}$ |
| 34 | $\frac{1}{2a^2} \sin at \sinh at$ | $\frac{s}{s^4 + 4a^4}$ |
| 35 | $\frac{1}{2a^3} (\sinh at - \sin at)$ | $\frac{1}{s^4 - a^4}$ |
| 36 | $\frac{1}{2a^2} (\cosh at - \cos at)$ | $\frac{s}{s^4 - a^4}$ |
| 37 | $(1 + a^2 t^2) \sin at - at \cos at$ | $\frac{8a^3 s^2}{(s^2 + a^2)^3}$ |
| 38 | $L_n(t) = \frac{e^t}{n!} \frac{d^n}{dt^n} (t^n e^{-t})$ | $\frac{1}{s} \left[\frac{s-1}{s} \right]^n$ |
| 39 | $\frac{1}{\sqrt{\pi t}} e^{at} (1 + 2at)$ | $\frac{s}{(s - a)^{\frac{3}{2}}}$ |
| 40 | $\frac{1}{2\sqrt{\pi t^3}} (e^{bt} - e^{at})$ | $\sqrt{s - a} - \sqrt{s - b}$ |

| | $f(t)$ | $F(s)$ |
|----|---|---|
| 41 | $\frac{1}{\sqrt{\pi t}} - ae^{a^2 t} \operatorname{erfc}(a\sqrt{t})$ | $\frac{1}{\sqrt{s+a}}$ |
| 42 | $\frac{1}{\sqrt{\pi t}} + ae^{a^2 t} \operatorname{erfc}(a\sqrt{t})$ | $\frac{\sqrt{s}}{s-a^2}$ |
| 43 | $\frac{1}{a} e^{a^2 t} \operatorname{erf}(a\sqrt{t})$ | $\frac{1}{\sqrt{s}(s-a^2)}$ |
| 44 | $e^{a^2 t} \operatorname{erfc}(a\sqrt{t})$ | $\frac{1}{\sqrt{s}(\sqrt{s}+a)}$ |
| 45 | $\frac{1}{\sqrt{b-a}} e^{-at} \operatorname{erf}(\sqrt{b-a}\sqrt{t})$ | $\frac{1}{(s+a)\sqrt{s+b}}$ |
| 46 | $e^{a^2 t} \frac{b}{a} \operatorname{erf}(a\sqrt{t}) - 1$ | $\frac{b^2 - a^2}{\sqrt{s}(s-a^2)(\sqrt{s}+b)}$ |
| 47 | $J_0(at)$ | $\frac{1}{\sqrt{s^2+a^2}}$ |
| 48 | $J_0(2\sqrt{kt})$ | $\frac{1}{s} e^{-k/s}$ |
| 49 | $\frac{1}{\sqrt{\pi t}} \cos 2\sqrt{kt}$ | $\frac{1}{\sqrt{s}} e^{-k/s}$ |
| 50 | $\frac{1}{\sqrt{\pi t}} \cosh 2\sqrt{kt}$ | $\frac{1}{\sqrt{s}} e^{k/s}$ |
| 51 | $\frac{1}{\sqrt{\pi k}} \sin 2\sqrt{kt}$ | $\frac{1}{s^{3/2}} e^{-k/s}$ |
| 52 | $\frac{1}{\sqrt{\pi k}} \sinh 2\sqrt{kt}$ | $\frac{1}{s^{3/2}} e^{k/s}$ |
| 53 | $\frac{k}{2\sqrt{\pi t^3}} \exp -\frac{k^2}{4t}$ | $e^{-k\sqrt{s}} \quad (k > 0)$ |

| | $f(t)$ | $F(s)$ |
|----|---|--|
| 54 | $\operatorname{erfc} \frac{k}{2\sqrt{t}}$ | $\frac{1}{s} e^{-k\sqrt{s}} (k \geq 0)$ |
| 55 | $\frac{1}{\sqrt{\pi t}} \exp -\frac{k^2}{4t}$ | $\frac{1}{\sqrt{s}} e^{-k\sqrt{s}} (k \geq 0)$ |
| 56 | $2\sqrt{\frac{t}{\pi}} \exp -\frac{k^2}{4t} - k \operatorname{erfc} \frac{k}{2\sqrt{t}}$ | $s^{-3/2} e^{-k\sqrt{s}} (k \geq 0)$ |
| 57 | $-e^{ak} e^{a^2 t} \operatorname{erfc} a\sqrt{t} + \frac{k}{2\sqrt{t}} + \operatorname{erfc} \frac{k}{2\sqrt{t}}$ | $\frac{ae^{-k\sqrt{s}}}{s(a + \sqrt{s})} (k \geq 0)$ |
| 58 | $e^{ak} e^{a^2 t} \operatorname{erfc} a\sqrt{t} + \frac{k}{2\sqrt{t}}$ | $\frac{e^{-k\sqrt{s}}}{\sqrt{s}(a + \sqrt{s})} (k \geq 0)$ |
| 59 | $\frac{1}{t}(e^{bt} - e^{at})$ | $\log \frac{s-a}{s-b}$ |
| 60 | $\frac{2}{t}(1 - \cos at)$ | $\log \frac{s^2 + a^2}{s^2}$ |
| 61 | $\frac{2}{t}(1 - \cosh at)$ | $\log \frac{s^2 - a^2}{s^2}$ |
| 62 | $\frac{1}{t} \sin kt$ | $\tan^{-1} \frac{k}{s}$ |

فصل ۸

تبدیل فوریه و پردازش سیگنال

۱.۸ مقدمه

در مدارهای الکترونیکی، اغلب با ولتاژ و جریان که نمایشگر سیگنال حامل اطلاعات مفید است، مواجه هستیم. مثلاً

- صوت، شامل موسیقی و صدا.
- تصویر، شامل تلویزون و نظارت.
- سنجش از دور حسگرهای دما، اندازه‌گیری کشش یا فشار، دستگاه‌های پزشکی، اندازه‌گیری توان، و چیزی‌هایی از این قبیل.
- سیگنال‌های رادار، لورن، جی‌پی‌اس و سیستمهای هدایت در هواپیماها و کشتی‌ها.
- سیگنال‌های مدوله شده مانند AM، UHF، VHF، FM^۱، CDMA^۲، GSM^۳ (دسترسی تقسیم کد) برای پخش همگانی و ارتباط نقطه به نقطه صوت، تصویر، صدا، و داده.

دلائل بیشماری برای پردازش سیگنال‌ها وجود دارد. برای سیگنال‌های صوتی و تصویری، ممکن است بخواهیم نویز را فیلتر کرده یا پهنه‌ای باند را تقلیل دهیم. برای سیگنال‌های سنجش از دور^۴ ممکن است بخواهیم داده‌های حسگرهای مختلف را آزمایش کرده تا به هدفهایی از قبیل پیشگوئی هوا، کم کردن سوخت خودرو، یا تفییر داده نمودار قلب دست‌یابیم. در حالت سیستمهای مخابراتی، می‌توانیم از مدولاسیون برای ارسال همزمان چندین سیگنال صوت و تصویر در یک محیط مشترک (مثل کابل هم‌محور، یا فیبر نوری) یا از درون فضای آزاد استفاده کنیم. بعلاوه، برای نمونه می‌خواهیم از افت مسیر و تداخل علائم^۵ که در کانال‌های مخابراتی پیش می‌آید اطلاع و ارزیابی داشته باشیم. ابزار ریاضیات اولیه که برای پردازش سیگنال‌ها بکار می‌رود تبدیل فوریه^۶ است. این تبدیل روشی برای تشریح سیگنال‌ها بر حسب مولفه‌های فرکانسی‌شان (یعنی، "حوزه فرکانس" بجای "حوزه زمان" که ما با آن ملموس و آشنا هستیم، تدارک می‌نماید. جدول (۱.۸) چندین نوع سیگنال و طیف فرکانسی‌های

^۱code division multiple access (CDMA)

^۲GSM (Groupe spéciale mobile)

^۳elementry

^۴intersymbol interference (ISI)

^۵Fourier transform

مربوطه (یعنی ردیف مولفه‌های فرکانسی که ممکن است در نمایش شکل موج باشند) آورده شده است. برای بقیه این فصل، ما سیگنال‌های صوتی را در مثال‌های خود بکار میبریم زیرا آنها برای خواندن‌گان آشناتر است.

جدول ۱.۸: سیگنال‌ها و طیف فرکانسی آنها

| نوع سیگنال | طیف فرکانسی آن |
|-----------------------------------|---|
| سخن‌گوئی | 3200Hz تا 200Hz |
| صوت با کیفیت عالی (طیف گوش انسان) | 20,000Hz تا 20Hz 1606.5kHz تا 526.6kHz |
| رادیو AM | 108.5MHz تا 87.5MHz |
| رادیو FM | 2.473GHz تا 2.401GHz |
| شبکه واي‌فاي networking | 0 تا ∞ هرتز |
| نویز سفید | 440Hz فرکانس اصلی |
| بالای وسط C در پیانو | 880Hz, 1320Hz, 1760Hz, ... هارمونیکها |

شکل (۱.۸) نمونه‌ای از شکل موج برخی از سیگنال‌های جدول (۱.۸) را نشان می‌دهد. بطورکلی، "تیزی" شکل هر یک از سیگنال‌ها مربوط به مولفه‌های فرکانسی‌ای زیاد آنها و ناهمواری و دندانه‌دار و عریض بودن طیف سیگنال دارد. بنابراین، بالای ۲۵ msec فواصل، سیگنال صوتی با کیفیت بالا در شکل (۱.۸)(ب) بیشتر از سیگنال سخن‌گوئی در شکل (۱.۸)(الف) دندانه‌دار است. سیگنال‌های متناوب (مثل نت‌های موسیقی در شکل (۱.۸)(ج)) شامل یک فرکانس اصلی بعلاوه مولفه‌های هارمونیکی است که باعث میشود شکل از حالت سینوسی بودن منحرف شود.

۲.۸ تشریح ریاضی سیگنال‌های متناوب: سری فوریه

با سیگنال x که ممکن است نمایشگر صدای قابل شنوایی که توسط مبدل (مثلاً میکروفون) بصورت ولتاژ است، باشد. اگر $x(t)$ مانند شکل (۱.۸)(ج)، یعنی نت A پیانو متناوب باشد، در این صورت نظریه فوریه بما میگوید که شکل موج را می‌توان بصورت جمع جملات سینوزوئیدال وزن شده‌ای بیان کرد:

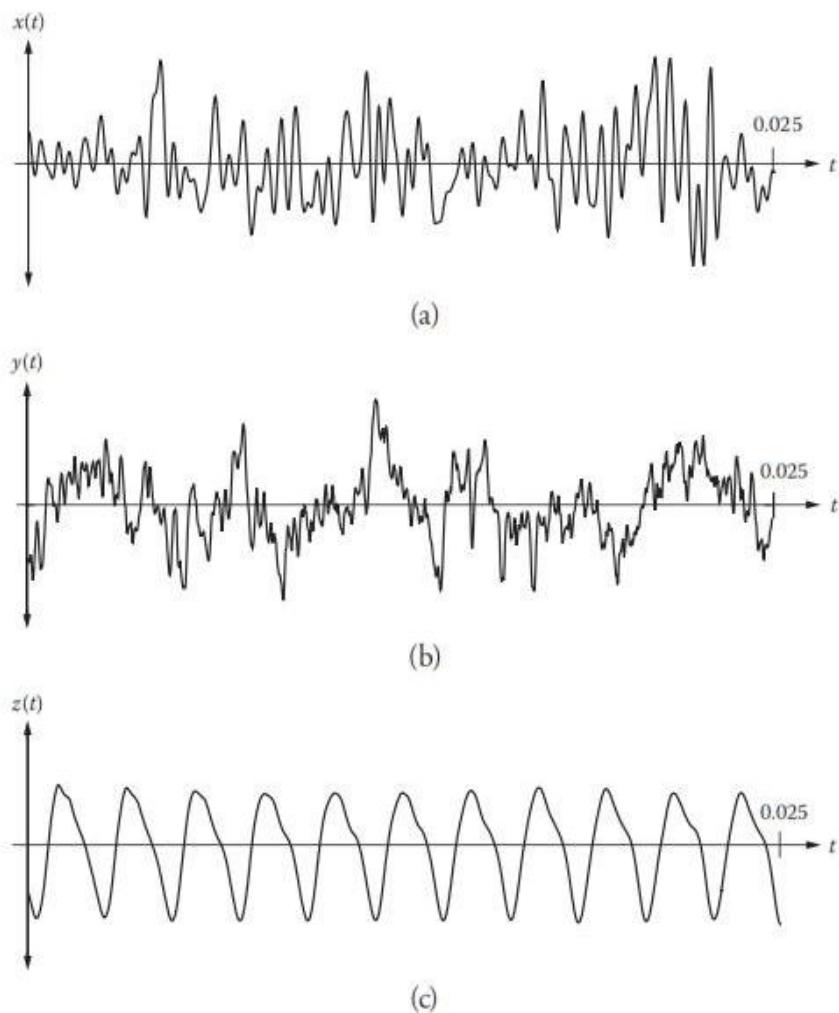
$$x(t) = A_0 + \sum_{n=1,2,3\dots} A_n \cos(2\pi n f_0 t) + \sum_{n=1,2,3\dots} B_n \sin(2\pi n f_0 t) \quad (۱.۸)$$

فرکانسی $n f_0$ در جملات دوم و سوم نمایشگر تمام مولفه‌های سینوسی و کسینوسی $x(t)$ است. بازاء $n = 1$ را فرکانس اصلی f_0 و بازاء $n > 1$ را هارمونیک آن میگویند. ضرایب A_n و B_n را ضرایب سری فوریه و A_0 را مولفه "DC" سیگنال $x(t)$ هستند. می‌توانیم A_0 ، A_n ، و B_n را با ضرب $x(t)$ در مولفه فرکانسی و سپس انتگرال گیری در یک دوره تناوب $x(t)$ تعیین کنیم. برای مثال، برای تعیین ضریب فوریه جمله m ام کسینوسی، دوطرف معادله (۱.۸) را در $\cos(2\pi m f_0 t)$ ضرب کرده و در یک دوره تناوب $T = \frac{1}{f_0}$ که در آن $\frac{T}{2} \leq t \leq -\frac{T}{2}$ است، انتگرال می‌گیریم:

$$\begin{aligned} & \int_{-T/2}^{T/2} x(t) \cos(2\pi n f_0 t) dt \\ &= \int_{-T/2}^{T/2} \left[A_0 + \sum_{n=1,2,3\dots} A_n \cos(2\pi n f_0 t) + \sum_{n=1,2,3\dots} B_n \sin(2\pi n f_0 t) \right] \cos(2\pi m f_0 t) dt \end{aligned} \quad (۲.۸)$$

چون توابع کسینوسی و سینوسی متعامد^۶ هستند، تمام جملات دست راست رابطه (۲.۸) بجز

^۶orthogonal



شکل ۱.۸:

صفر خواهند شد. یعنی،

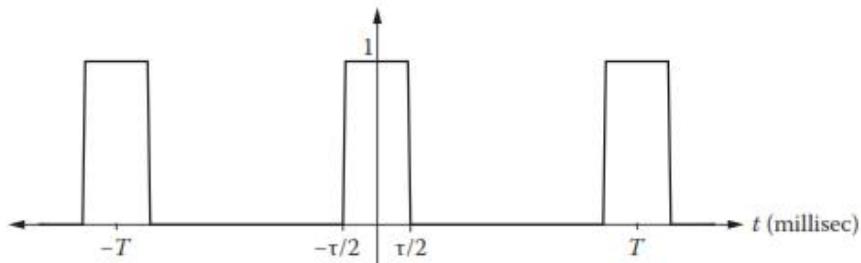
$$\begin{aligned}
 \int_{-T/2}^{T/2} \cos(2\pi n f_0 t) \cos(2\pi m f_0 t) dt &= \frac{T}{2} & m = n \\
 \int_{-T/2}^{T/2} \cos(2\pi n f_0 t) \cos(2\pi m f_0 t) dt &= 0 & m \neq n \\
 \int_{-T/2}^{T/2} \cos(2\pi m f_0 t) \sin(2\pi n f_0 t) dt &= 0 & \text{تمام}
 \end{aligned} \tag{۳.۸}$$

بنابراین، عمل انتگرال‌گیری تمام مولفه‌های $x(t)$ را بجز برای فرکانس $m = n$ "فیلتر" می‌کند. با اعمال

رابطه (۴.۸) به رابطه (۲.۸) خواهیم داشت

$$\begin{aligned} A_n &= \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cos(2\pi n f_0 t) dt \\ B_n &= \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \sin(2\pi n f_0 t) dt \end{aligned} \quad (4.8)$$

برای محاسبه A_0 ضریب DC , بازاء $m = 0$ را انتگرال می‌گیریم و خواهیم داشت



شکل ۲.۸: یک قطار پالس که در فواصل زمانی τ "روشن" و زمان تناوب آن T است.

$$A_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) dt \quad (5.8)$$

مثال ۱.۸:

سری فوریه را برای پالس متناوب شکل (۲.۸) با زمان تناوب T , فرکانش $f_0 = \frac{1}{T}$ که در زمان τ "روشن" و زمان $T - \tau$ "خاموش" است, محاسبه کنید.
سیگنال $x(t)$ را روی محور y ها مرکزیت داده تا اطمینان حاصل شود که $x(t)$ یک سیگنال زوج است. بنابراین, بعلت تقارن, تمام جملات فرد رابطه (۱.۸) (یعنی, جملات سینوسی) صفر خواهند بود. ابتداء با جمله DC (برای $n = 0$ رابطه (۵.۸) شروع می‌کنیم. از $x(t)$ برای یک دوره تناوب انتگرال می‌گیریم:

$$A_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (1) dt = \frac{\tau}{T}$$

برای جملات بالاتر داریم:

$$\begin{aligned} A_n &= \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (1) \cos(2\pi n f_0 t) dt \\ &= \frac{2}{T} \left[\frac{\sin(2\pi n f_0 t)}{2\pi n f_0} \right]_{-\frac{T}{2}}^{\frac{T}{2}} \\ &= \frac{2}{\pi n} \sin(\pi n f_0 \tau) \end{aligned}$$

بنابراین, جواب برای چند جمله اول سری فوریه برای قطار پالس خواهد بود

$$x(t) = \frac{\tau}{T} + \frac{2}{\pi} \sin\left(\frac{\pi\tau}{T}\right) \sin(2\pi f_0 t) + \frac{1}{\pi} \sin\left(\frac{4\pi\tau}{T}\right) \sin(4\pi f_0 t) + \frac{2}{3\pi} \sin\left(\frac{6\pi\tau}{T}\right) \sin(6\pi f_0 t) + \dots$$

برنامه متلب زیر ضرایب سری فوریه را برای قطار پالس بازاء $T = 1 \text{ ms}$ و $\tau = 0.25 \text{ ms}$ محاسبه کرده و قطار پالس را بر اساس ضرایب سری فوریه بازسازی کنید.

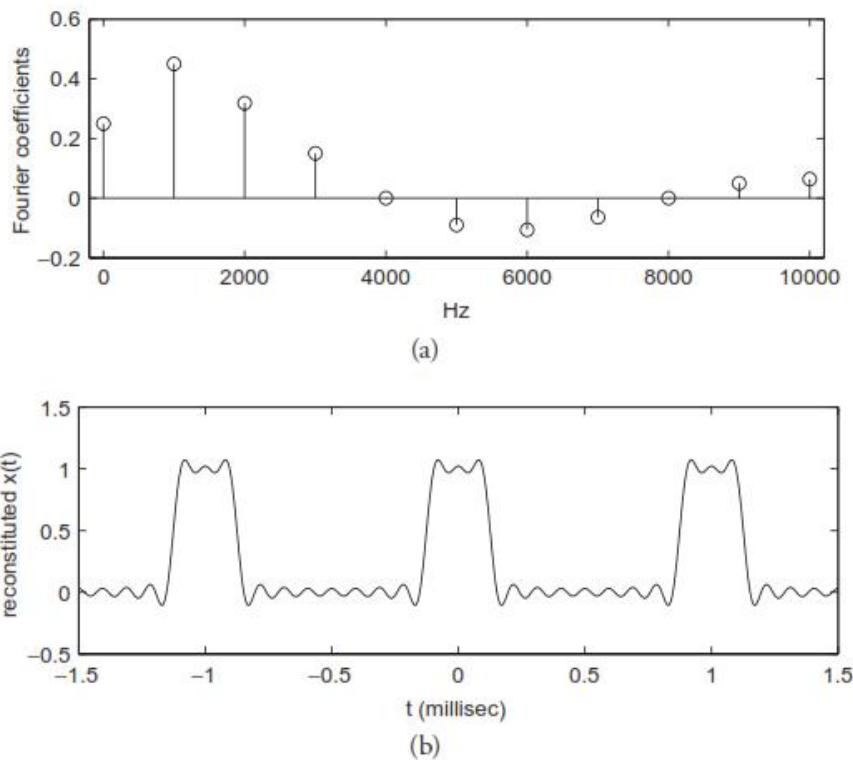
```
% Example_8_1.m
% Compute the Fourier series for a pulse train.
T=1e-3; % pulse period
fo=1/T; % frequency of pulse train in Hz
tau=0.25e-3; % "on" time of a single pulse
n=1:10; % we will compute the first 10 coefficients
% compute the DC coefficient:
A0 = tau/T;
% compute coefficients 1 thru n
for i=1:length(n)
    An(i) = (2/(pi *n(i))) * sin(pi*n(i)*fo*tau);
end
% Plot the coefficients:
subplot(2,1,1);
stem([0 fo*n],[A0 An]);
xlabel('Hz'); ylabel('Fourier coefficients');
axis([-200 10200 -.2 .6]);
% Use the Fourier coefficients to reconstitute the pulse
% train over the interval -1.5 <t <1.5 millisecond with n
% coefficients
t=-1.5e-3:1e-6:1.5e-3;
waveform = zeros(1,length(t));
% Add in the DC coefficient
waveform = waveform + A0;
% Add in signal components for 1 thru nth coefficients
for i=1:length(n)
    waveform = waveform + An(i) * cos(2*pi*n(i)*fo*t);
end
subplot(2,1,2);
plot(t*1000,waveform);
xlabel('t (millisecond)'); ylabel('reconstructed x(t)');
%-----
```

ضرایب ده جمله اول سری فوریه محاسبه شده در شکل (۳.۸)(الف) و شکل بازسازی شده قطار پالس بازه ضرایب ده جمله در شکل (۳.۸)(ب) نشان داده شده است.

۳.۸ سری نمائی فوریه مختلط و تبدیل فوریه

عموماً، سری فوریه می‌تواند ترکیبی از جمع هر مجموعه توابع متعامد وزن شده باشد. با وجودی که سینوس و کسینوس بعلت آشنائی ما با آنها انتخاب عمومی است، از دید ریاضیات مجموعه توابع نمائی مختلط انتخاب برتری بحساب می‌آید:

$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_0 t} \quad (4.8)$$



شکل ۳.۸: (الف) ضرایب فوریه برای قطار پالس شکل (۲.۸) بازه بازه $T = 1 \text{ ms}$ و $\tau = 0.25 \text{ ms}$. (ب) قطار پالس بازسازی شده بازه اول سری فوریه. توجه کنید که حذف جملات بالاتر دلیل عدم تیز بودن مخصوصاً لبه ها میشود.

از اتحادها میدانیم $\sin x = \frac{e^{jx} - e^{-jx}}{2j}$ و $\cos x = \frac{e^{jx} + e^{-jx}}{2}$ می توان ثابت کرد که رابطه (۶.۸) معادل رابطه (۱.۸) است و در آن $C_0 = A_0$, $C_{-n} = \frac{A_n + jB_n}{2}$, $C_n = \frac{A_n - jB_n}{2}$ میباشد. با ضرب یک تابع مزدوج مختلط نمائی و انتگرال گیری در بازه یک دوره تناوب رابطه متعامد بودن آنها را می توان اثبات کرد:

$$\int_{-T/2}^{T/2} e^{j2\pi m f_0 t} e^{-j2\pi n f_0 t} dt = \begin{cases} T & m = n \\ 0 & m \neq n \end{cases} \quad (7.8)$$

انتگرال گیری از دو طرف معادله (۶.۸) در بازه یک دوره تناوب و اعمال رابطه (۷.۸) رابطه ای برای ضرایب سری فوریه نمائی مختلط برای هر سیگنال متناوب مانند $x(t)$ میدهد:

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j2\pi n f_0 t} dt \quad (8.8)$$

اکنون دو اصلاحیه در رابطه (۸.۸) بکار میبریم. اول، ضرایب سری فوریه را با تغییر دادن عامل $\frac{1}{T}$ از سمت چپ به راست نرماییزه می کنیم. سپس، ما $X(f)$ را عنوان ضرایب فوریه نرماییزه نشان میدهیم. رابطه (۸.۸) خواهد شد

$$X(n f_0) = T C_n = \int_{-T/2}^{T/2} x(t) e^{-j2\pi n f_0 t} dt \quad (9.8)$$

بعداً، بجای $n f_0$ متغیر f (برای فرکانس) جایگزین و $T \rightarrow \infty$ میل میدهیم از اینرو $x(t)$ یک سیگنال

غیرتناوبی^v میشود. در این صورت، حدود انتگرال در رابطه (۹.۸) بینهایت شده و $X(f)$ یک تابع پیوسته خواهد شد:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi f t} dt \quad (10.8)$$

رابطه (۱۰.۸) را تبدیل فوریه^۸ تابع زمانی $x(t)$ تعریف می‌کنیم.

مثال ۲.۸

برای قطار پالس شکل (۲.۸) ضرایب فوریه نرمالیزه را رسم کنید. عرض پالس را $\tau = 0.25 \text{ msec}$ و سه نمودار بازاء $T = 1, 2, 10 \text{ msec}$ رسم کنید.

همانطور که برای مثال ۱.۸ انجام شد، مولفه DC برابر $C_0 = \frac{\tau}{T}$ است. برای جملات $n \neq 0$ داریم

$$\begin{aligned} C_n &= \frac{1}{T} \int_{-\tau/2}^{\tau/2} (1) \cdot e^{-j2\pi n f_0 t} dt \\ &= \frac{1}{T} \left[\frac{e^{-j2\pi n f_0 t}}{-j2\pi n f_0} \right]_{-\tau/2}^{\tau/2} \\ &= \frac{\sin \pi n f_0 \tau}{\pi n} \end{aligned}$$

برنامه‌ای که ضرایب را محاسبه و رسم می‌کند بقرار زیر است:

```
% Example_8_2.m
% Compute normalized exponential Fourier series
% coefficients for a pulse train
tau = 0.25e-3; % "on" time of a single pulse
T = [ 1e-3 2e-3 10e-3 ]; % pulse train periods
fo = 1 ./ T; % frequency of pulse train in Hz
n = -150:150;
for i=1:length(T)
    Cn(i,:) = ( 1 ./ (pi*n) ) .* sin(pi*n*fo(i)*tau);
% Find the "zero" entry in n and compute it
% separately (because it has a different formula than
% the rest):
    zeroeth_element = find(n==0);
    Cn(i,zeroeth_element) = tau/T(i);
% Normalize the coefficients:
    Cn_normalized(i,:) = Cn(i,:) .* T(i);
end
% find reasonable limits for axes so that all 3 plots
% have the same min/max
x_axis_limits = [ -3.8/tau 3.8/tau ];
y_axis_limits = [ 1.5*min(Cn_normalized(1,:)) ...
    1.25*max(Cn_normalized(1,:)) ];
% Plot separately for each value of T:
subplot(3,1,1);
```

^vaperiodic signal

^۸Fourier transform

```

stem( n*fo(1),Cn_normalized(1,:) );
ylabel({'Fourier coefficients';'for T=1 ms'});
axis([x_axis_limits y_axis_limits]);
title('Exponential Fourier Series for Pulse');
subplot(3,1,2);
stem( n*fo(2),Cn_normalized(2,:) );
axis([x_axis_limits y_axis_limits]);
ylabel({'Fourier coefficients';'for T=2 ms'});
subplot(3,1,3);
plot( n*fo(3),Cn_normalized(3,:) );
axis([x_axis_limits y_axis_limits]);
xlabel('f (Hz)');
%-----

```

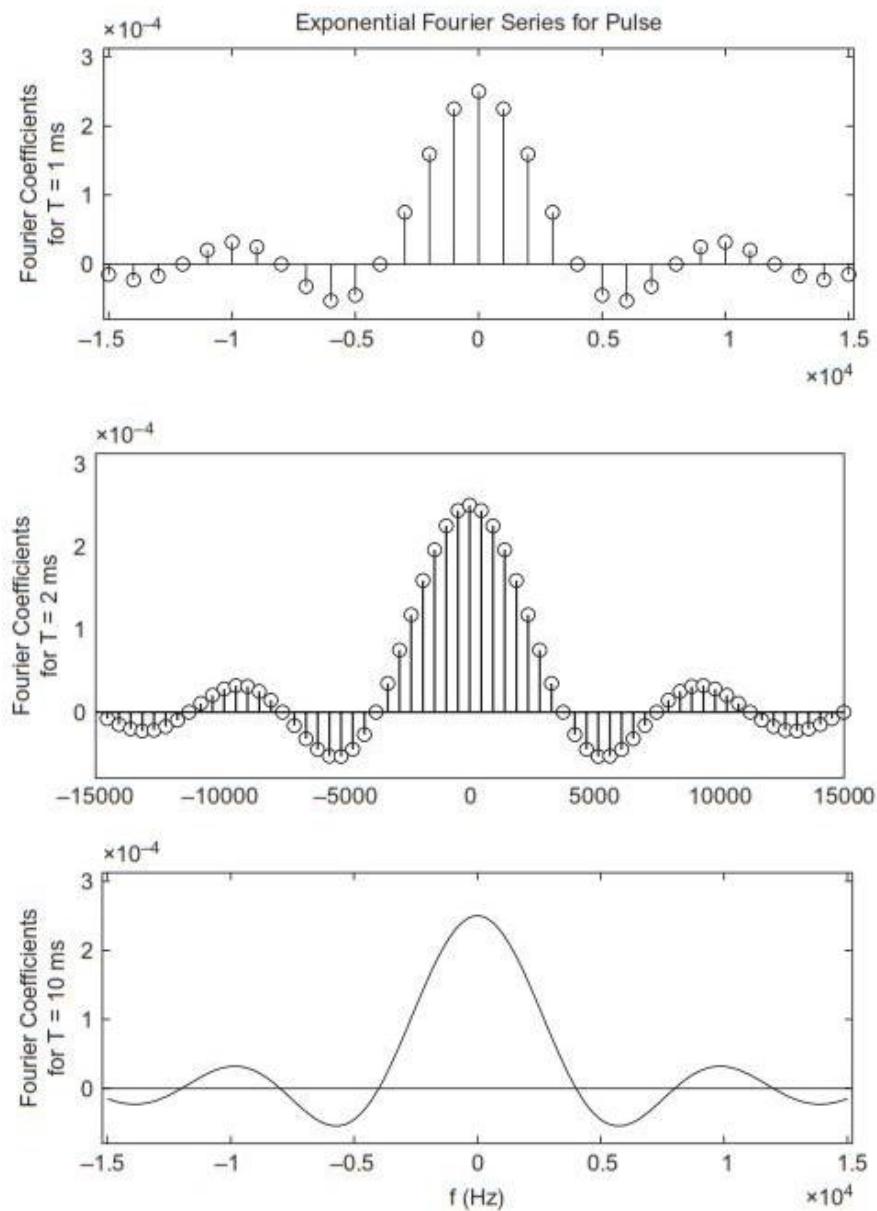
شکل (۴.۸) سه نمودار ضرایب سری فوریه را که در مثال ۲.۸ محاسبه شده است برای $T = 1, 2, 10 \text{ msec}$ نشان میدهد. توجه کنید که وقتی عرض پالس ثابت است، با افزایش زمان تناب، ضرایب سری نمائی فوریه بهم نزدیکتر و نزدیکتر می‌شود. وقتی $T = 10 \text{ msec}$ است، آنها آنقدر بهم نزدیک هستند که شبیه یکتابع پیوسته رسم می‌شوند. در حد $\infty \rightarrow T$ ، شکل (۴.۸)(ج) مربوط به تبدیل فوریه پالس می‌شود.

۴.۸ خواص تبدیل فوریه

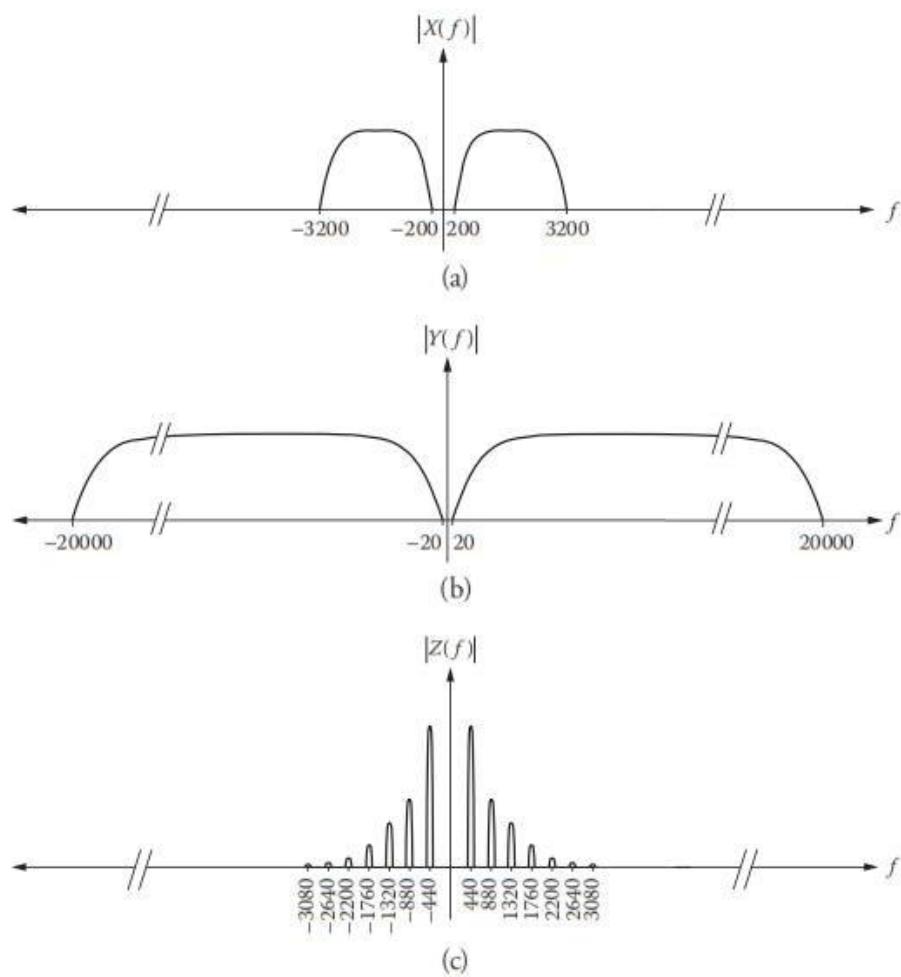
از نظر کیفی، تبدیل فوریه توصیف فرکانسی سیگنال $x(t)$ است. شکل (۵.۸) سه سیگنال شکل (۱.۸) در حوزه فرکانس نشان میدهد. توجه کنید که وقتی خواص آنها توصیف می‌شود، شکل واقعی تبدیل فوریه‌ها بطور کلی غیر یکنواخت هستند (و در این حالت ما آنها را بصورت مستطیل‌های گرد شده رسم کرده‌ایم). در حال حاضر، نقاط ابتداء و انتهائی برای ما خیلی مهم است. توجه کنید که تبدیلات فوریه همواره حول محور y ها متقارن هستند، زیرا همواره فرض می‌کنیم که $x(t)$ از یک شکل موج واقعی که قابل تحقق و حقیقی است سرچشمه گرفته است (و از این‌رو مولفه موهومی ندارد). با این وجود، تبدیلات فوریه دارای مولفه موهومی بوده، و در شکل (۵.۸)، ما فقط مقدار (قدرمطلق) آنرا رسم کرده‌ایم (نه فاز آنرا). ایده یک سیگنال که دارای مختلط یا فرکانس منفی باشد ممکن است گیج کننده بنظر برسد، اما آن محصول مصنوعی استفاده از تابع نمائی مختلط بعنوان توابع متعامد است. وقتی لازم باشد، می‌توانیم فرکانس مختلط را فرکانس حقیقی با اعمال $e^{jx} = \cos x + j \sin x$ تغییر دهیم؛ در همان وقت، می‌توانیم از ریاضیات ساده سازی استفاده کرده و جملات سینوس و کسینوس را بصورت یک جمله نمائی ترکیب کنیم.

از نظر ریاضی، تبدیل فوریه را ممکن است زیرمجموعه تبدیل لاپلاس دوطرفه که در آن $s = 2\pi f$ است در نظر گرفت [۴]. البته آن تحلیل کاملی مانند تبدیل لاپلاس انجام نمی‌دهد (یعنی نمی‌توان آنرا برای پیشگوئی پایداری سیستم استفاده کرد)، اما سادگی آن رفتار سیگنال را براحتی در جلو دیدگان ما قرار میدهد. مانند هر روش تبدیلی، می‌توان از یک حوزه به حوزه دیگر رفت و هر کدام ساده‌تر باشد از آنها استفاده کرد.

می‌توانیم تبدیل فوریه معکوس را با خارج کردن تبدیل فوریه از حالت نرمالیزه $\frac{X(f)}{T}$ و برگشت



شکل ۴.۸: نمودار ضرایب سری فوریه برای یک قطاع پالس به عرض $msec = 0.25\tau$. (الف) برای زمان تناوب یک میلی ثانیه، (ب) برای دو میلی ثانیه، (ج) برای ده میلی ثانیه. برای حالت آخر، ضرایب بهم نزدیک شده که آنها بصورتتابع پیوسته رسم شده است.



شکل ۵.۸: تبدیل فوریه برای چند نوع سیگنال صوتی: (الف) سخن‌گوئی؛ (ب) صوت با کیفیت بالا؛ (ج) نت بالای سطح C پیانو.

جدول ۲.۸: تبدیل فوریه توابع معروف

| خواص | حوزه زمان | $x(t)$ | حوزه فرکانس |
|-------------------------|--|--------|---|
| خطی بودن | $x(t) = Ay(t) + Bz(t)$ | | $X(f) = AY(f) + B(Z(f))$ |
| دوگانگی | $X(t)$ | | $x(-f)$ |
| ضربه | $x(t) = \delta(t)$ | | $X(f) = 1$ |
| کسینوس | $x(t) = \cos 2\pi f_0 t$ | | $X(f) = \frac{\delta(f-f_0)+\delta(f+f_0)}{2}$ |
| سینوس | $x(t) = \sin 2\pi f_0 t$ | | $X(f) = \frac{\delta(f-f_0)-\delta(f+f_0)}{2j}$ |
| پالس | $x(t) = \begin{cases} 1 & t \leq \tau \\ 0 & \text{در غیر این صورت} \end{cases}$ | | $X(f) = \frac{\sin 2\pi f \tau}{\pi f}$ |
| فیلتر پائین‌گذر ایده‌آل | $x(t) = \frac{\sin 2\pi f_0 t}{\pi t}$ | | $X(f) = \begin{cases} 1 & f \leq f_0 \\ 0 & \text{در غیر این صورت} \end{cases}$ |
| کانولوشن | $y(t) = x(t) * h(t)$ | | $Y(f) = X(f)H(f)$ |
| تابع نمائی میراوشونده | $y(t) = e^{-\alpha t} u(t)$ | | $Y(f) = \frac{1}{\alpha + j2\pi f}$ |
| کسینوس میراوشونده علی | $y(t) = e^{-at} \cos(2\pi f_0 t) u(t)$ | | $Y(f) = \frac{a + j2\pi f}{(a + j2\pi f)^2 + (2\pi f_0)^2}$ |
| سینوس میراوشونده علی | $y(t) = e^{-at} \sin(2\pi f_0 t) u(t)$ | | $Y(f) = \frac{2\pi f_0}{(a + j2\pi f)^2 + (2\pi f_0)^2}$ |

به عقب رابطه (۴.۸) بدست آورد

$$\begin{aligned} x(t) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} X(nf_0) e^{j2\pi n f_0 t} \\ &= \sum_{n=-\infty}^{\infty} X(nf_0) e^{j2\pi n f_0 t} \frac{nf_0}{n} \end{aligned} \quad (11.8)$$

با قرار دادن $f/n \rightarrow df$ ، در حد جمع به انتگرال تبدیل می‌شود:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} df \quad (12.8)$$

رابطه (۱۲.۸) تعریف تبدیل فوریه معکوس $X(f)$ است.

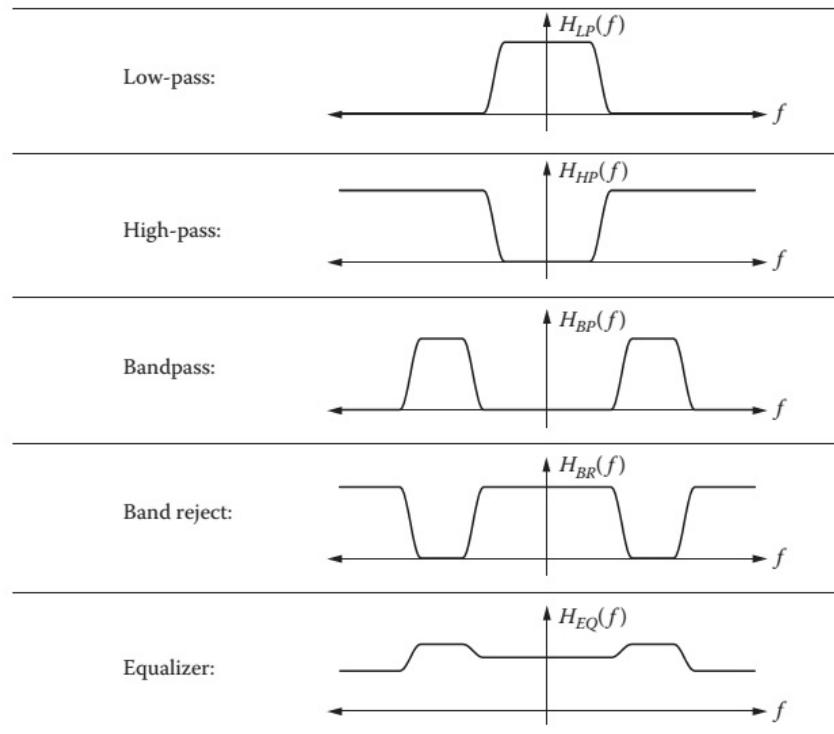
برخی از خواص مهم تبدیل فوریه در جدول (۲.۸) خلاصه شده است. به شباهت تبدیل فوریه مستقیم و تبدیل فوریه معکوس (رابطه ۱۲.۸ و ۱۲.۸) توجه کنید. تفاوت آنها فقط در یک علامت منفی نمائی است. این منجر به خاصیت دوگانگی^۹ می‌شود که در جدول (۲.۸) می‌توان آنرا مشاهده کرد. برای مثال، پالسی با عرض 2τ در حوزه زمان، دارای تبدیل فوریه $\frac{\sin 2\pi f \tau}{\pi f}$ در حوزه فرکانس بوده و می‌توانیم نتیجه بگیریم که یک "پالس" به عرض $2F_0$ در حوزه فرکانس دارا تبدیل فوریه معکوس $\frac{\sin 2\pi F_0 t}{\pi F_0}$ در حوزه زمان است.

۵.۸ فیلترها

فیلتر کردن کاربرد تابع و استه به فرکانس روی یک شکل موج است. انواع مختلف فیلترها طبقه بندی شده و با شکل پاسخ فرکانسی در حوزه فرکانس مشخص می‌شود. آنها در جدول (۳.۸) خلاصه شده‌اند.

^۹duality property

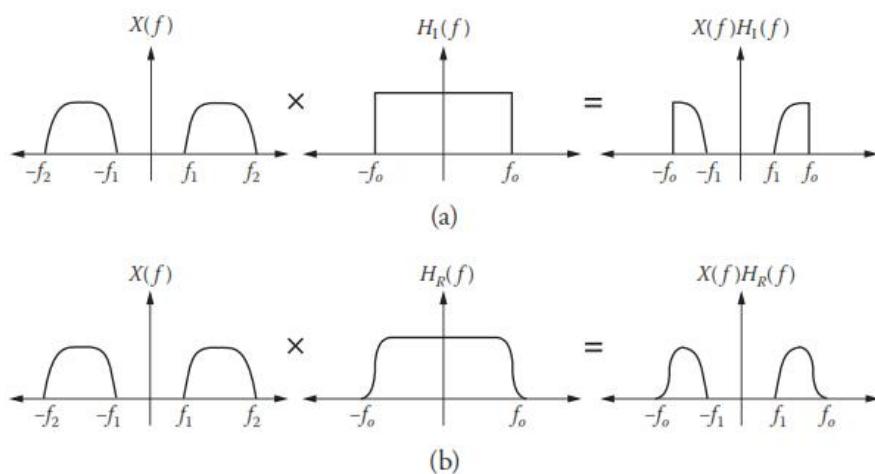
جدول ۳.۸: پاسخ فرکانسی برای فیلترهای معروف



برای مثال، یک فیلتر پائین‌گذر^{۱۰} دارای مقدار واحد در فرکانس‌های پائین و مقدار صفر (یا نزدیک صفر) در فرکانس‌های بالا است. در این حالت، ناحیه فرکانس‌های پائین را باند عبور^{۱۱} می‌نامند زیرا سیگنال‌های بدون تغییر از درون آن می‌گذرند، و ناحیه فرکانس‌های بالا را باند توقف زیرا سیگنال‌ها در این ناحیه بوسیله فیلتر متوقف می‌شوند. برعکس، فیلترهای بالاگذر^{۱۲} در فرکانس‌های پائین دارای باند توقف و در فرکانس‌های بالا دارای باند عبور هستند. در فیلترهای میان‌گذر^{۱۳} دارای باند عبور محدود در فرکانس‌های بالا و پائین هستند. فیلترهای میان‌گذر^{۱۴} (بر عکس فیلتر میان‌گذر) دارای باند توقف در فرکانس‌های بالا و پائین است. یک متعادل کننده^{۱۵} دارای چندین باند عبور با مقدار متغیر است و عملأً برای ترمیم رفتار فرکانس غیرخطی یک تقویت کننده یا کانال‌های مخابراتی است.

شکل (الف)(۶.۸) مثالی از یک فیلتر پائین‌گذر با فرکانس قطع f_0 و پاسخ فرکانسی $H_I(f)$ مربوطه که در آن اندیس I نمایشگر "ایده‌آل" است) اگر این فیلتر بر یک سیگنال با تبدیل فوریه $X(f)$ که شامل فرکانس‌های $f_1 < |f| < f_2$ هستند، در این صورت خروجی فیلتر حاصل ضرب $H_I(f)X(f)$ و

^{۱۰}low pass filter^{۱۱}passband^{۱۲}high pass filter^{۱۳}bandpass filter^{۱۴}band reject filter^{۱۵}equalizer



شکل ۸.۶: (الف) یک فیلتر "دیوارآجری" پائین‌گذر با فرکانس قطع f_0 . (ب) یک فیلتر پائین‌گذر قابل تحقق لزوماً انتقال باند ملایمی دارد.

تمام فرکانس‌های بیش از f_0 حذف می‌شود.

متاسفانه، یک مسئله اساسی این فیلتر ایده‌آل "دیوارآجری" در حوزه فرکانس دارد. با توجه به جدول (۲.۸)، میدانیم که ضرب در حوزه فرکانس معادل کانولوشن در حوزه زمان است. اگر $H_I(f)$ را در حوزه زمان مشاهده کنیم، می‌بینیم که تابع $h_I(t) = \frac{\sin 2\pi f_0 t}{\pi t}$ مربوطه برای تمام زمانها $\pm t$ گسترش یافته است. بنابراین، برای کانولوشن $x(t)$ با $h_I(t)$ ، تمام مقادیر $x(t)$ از آن جمله $0 < t < T$ را نیاز داریم. فیلتری که وابسته به مقادیر زمانهای $0 < t < T$ باشد را **غیر علی مینامند** و ساخت آن از نظر عناصر مداری و شبیه سازی عددی غیر ممکن است.

با این فرض که نمی‌توانیم یک فیلتر ایده‌آل بسازیم، راه‌های دیگری وجود دارد که میتوانیم بجای آن بکار بردیم. یکی از آنها اعمال تابع پنجره^{۱۶} بر $h_I(t)$ است تا آنرا علی کند. ساده تراین تابع پنجره ضرب $\frac{\sin 2\pi f_0 t}{\pi t}$ در یک تابع پنجره مستطیلی است که بخش زمانهای منفی $h_I(t)$ را حذف کند و از بین ببرد، بنابراین آنرا مجبور به علی بودن کند. روش دیگر استفاده از مشخصه فیلتر های قابل تحقق فیزیکی است از قبیل پاسخ باترورث^{۱۷} و چهی چف^{۱۸} است. در هر حالت، نتیجه آخر حذف ناپیوستگی در پاسخ فرکانسی در فرکانس f_0 است. تابع $H_R(f)$ مربوطه (که در آن R بمعنی "قابل تحقیق"^{۱۹}) یک فیلتر ملایمی است که عملاً میتوان آنرا ساخت. این در شکل (۶.۸)(ب) نشان داده شده است. در اینجا (۶.۸) اکنون یک لبه خمیده است و خروجی فیلتر حاصل همان لبه را نیز دارد.

۶.۸ نمایش زمان گسسته سیگنالهای زمان پیوسته

در پردازش سیگنالهای دیجیتالی^{۲۰} (و در مطلب)، ما سیگنالهای زمان پیوسته را بصورت رشته‌ای از اعداد با فرآیند نمونه‌برداری^{۲۱} نمایش میدهیم. شکل (۶.۸)(الف) نمونه‌ای از یک سیگنال زمان

^{۱۶}window function

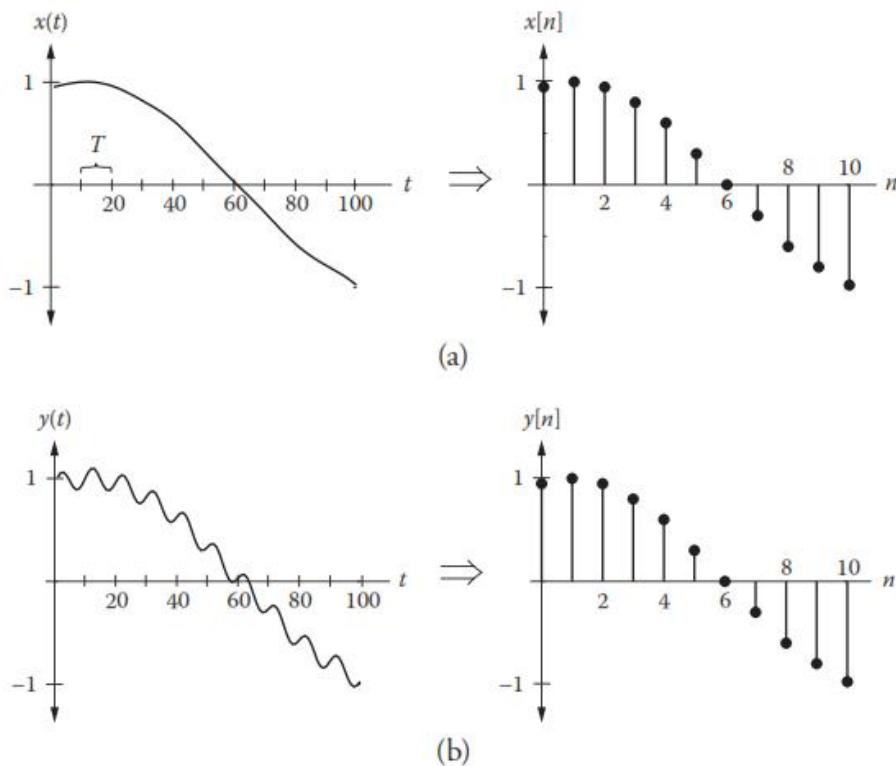
^{۱۷}Butterworth

^{۱۸}Chebyshev

^{۱۹}realizable

^{۲۰}digital signal processing

^{۲۱}sampling



شکل ۷.۸: (الف) نمونه‌ای از یک سیگنال زمان پیوسته $x(t)$ و سیگنال زمان گسسته $x[n]$ مربوط به آن، که با نمونه‌برداری از $x(t)$ در فواصل زمانی T بدست آمده است. (ب) اگر (y(t) دارای مولفه‌های فرکانسی بالائی که دقیقاً کسری از T باشد، داشته باشد، در آن صورت مولفه‌ها در $[n]$ مربوطه قابل مشاهده نخواهند بود.

پیوسته مانند $x(t)$ را (که آرگومان زمان در پرانتر است) به همراه سیگنال زمان گسسته مربوط به آن $x[n]$ در اینجا آرگومان گسسته در برآخت است) را نشان میدهد. ما $x[n]$ را از $x(t)$ با استخراج مقادیر $x(t)$ در فواصل زمانی معین T که زمان نمونه‌برداری ^{۲۲} (یا معادل آن نوخ نمونه‌برداری ^{۲۳} $f_S = \frac{1}{T}$) نامیده می‌شود. از نظر ریاضی، ما سیگنال را از زمان پیوسته به زمان گسسته با فرمول تبدیل می‌کنیم

$$x[n] = x(nT) \quad (13.8)$$

که در آن n عدد صحیح است. توجه کنید که سیگنال گسسته $x[n]$ هیچ اطلاعاتی از زمان همراه ندارد و فقط یک سری مقدار بازه اعداد صحیح n است. بنابراین، وقتی یک سیگنال زمان پیوسته را نمونه‌برداری می‌کنیم، مهم و لازم است که زمان نمونه‌برداری T را بخارط داشته باشیم زیرا برای تبدیل معکوس زمان گسسته به پیوسته الزامی است.

ضمناً باید سرعت (نوخ) نمونه‌برداری مناسب را در نظر گیریم. شکل (۱۳.۸)(ب) وقتی T را بزرگ در نظر گیریم، چه چیز میتواند اتفاق بیافتد را نشان می‌دهد. سیگنال گسسته مربوطه $y[n]$ مولفه سینوسی فرکانسی‌های زیاد (دقیقاً با فرکانس $\frac{1}{T}$) که روی $y(t)$ "سوار" است را حذف می‌کند. بنابراین، نمونه‌برداری سیگنال زمان پیوسته، یگانگی را در حوزه گسسته تضمین نمی‌کند و این در شکل (۱۳.۸) که $x[n]$ و $y[n]$ یکسان هستند واضح است. قضیه نمونه‌برداری می‌گوید که تشریح کامل یک سیگنال

^{۲۲}sampling time

^{۲۳}sampling rate

زمان گسسته بوسیله سیگنال زمان گسسته، باید سرعت نمونه برداری را حداقل دو برابر بیشترین مولفه فرکانسی سیگنال اصلی انتخاب کنیم. در عمل، اغلب سرعت نمونه برداری را کمی بیش از کمترین فرکانس لازم برای کامل نبودن مدارها و تغییر عناصر انتخاب می‌کنیم. برای مثال، در سی دی های صوتی^{۲۴}، که می‌توانند تا فرکانس‌های ۲۰,۰۰۰ Hz را تولید کند، سرعت نمونه برداری را ۴۴, ۱۰۰ Hz نمونه در ثانیه (بجای ۴۰,۰۰۰ Hz نمونه در ثانیه) انتخاب می‌نماییم. انگیزه‌های زیادی برای تبدیل سیگنال زمان پیوسته به زمان گسسته وجود دارد، اما دلیل مهم ما استفاده از کامپیوتر برای پردازش سیگنال با برنامه‌های شبیه مطلب است.

۷.۸ تبدیل فوریه سیگنال‌های زمان گسسته

تبدیل فوریه سیگنال زمان پیوسته^{۲۵} (که از این به بعد آنرا CFT مینامیم) که در بخش ۳.۸ معرفی شد را می‌توان برای سیگنال‌های زمان گسسته بعنوان تبدیل فوریه زمان گسسته^{۲۶} یا DTFT(ϕ) چنین تعریف کرد

$$X(\phi) = \sum_{n=-\infty}^{\infty} x[n] e^{-j2\pi n\phi} \quad (14.8)$$

و عکس DTFT مربوطه از رابطه زیر محاسبه می‌شود

$$x[n] = \int_{-1/2}^{1/2} X(\phi) e^{j2\pi n\phi} d\phi \quad (15.8)$$

توجه کنید که DTFT در رابطه (۱۴.۸) بصورت جمع (بجای انتگرال) تعریف می‌شود زیرا ماهیت $x[n]$ گسسته است. بعلاوه، چون $x[n]$ بدون دیمانسیون است، $X(\phi)$ مربوطه نیز بدون دیمانسیون است. رابطه (۱۴.۸) نسبت به ϕ بعلت جمله $e^{-j2\pi n\phi}$ متناوب است؛ بنابراین، بجای محاسبه تمام فرکانسها، فقط کافی است که $X(\phi)$ را در بازه $-\frac{1}{2} \leq \phi \leq \frac{1}{2}$ برای CFT محاسبه کنیم. اگر $x[n]$ از سیگنال زمان پیوسته $x(t)$ با سرعت f_s نمونه برداری شود، در آن صورت ϕ مربوط به فرکانس‌های $-\frac{f_s}{2} \leq f \leq \frac{f_s}{2}$ خواهد بود که تمام فرکانس‌های مفید را با توجه به نمونه برداری میدهد.

خیلی مهم است که در کنیم DTFT یکتابع پیوسته است نه گسسته. بعنوان مثال، فرض کنید یک موج سینوسی بفرکانس $1000Hz$ با 44100 نمونه در ثانیه نمونه برداری می‌شود. این بدین معنی است که نوع گسسته مربوطه موج سینوسی $1000Hz$ هر 44.1 نمونه تکرار شده، و $X(\phi)$ مربوطه مولفه‌های غیر صفر را در $\phi = \pm \frac{1000}{44100}$ دارد، که واضح است تمام اعداد نیستند. اگر $x[n]$ شامل مولفه‌های فرکانسی مختلفی باشد، در این صورت آنها را می‌توان در نسبت مربوطه f_s با $X(\phi)$ نمایش داده و یک تابع پیوسته تولید خواهد شد.

چون (۱۴.۸) نسبت به ϕ متناوب است، مقادیر $X(\phi)$ برای $\frac{1}{2} < |\phi| \leq \frac{1}{2}$ صرفاً گپی مقادیر ناحیه $-\frac{1}{2} \leq \phi \leq \frac{1}{2}$ است. دلیل آن است که $x[n]$ را می‌توان بصورت یک تابع متناوب در نظر گرفته که بوسیله سری فوریه با نمونه برداری از $X(\phi)$ می‌توان ضرایب را بدست آورد. این کاربرد معکوس همان روشی است که در بخش ۲.۸ برای بدست آوردن CFT بصورت "آلودن" ضرایب سری فوریه تابع پیوسته و متناوب، بکاربرده شد. از نظر ریاضی، "غیرآلودن" DTFT به ضرایب فوریه‌اش را تبدیل فوریه گسسته^{۲۷} یا DFT می‌نامند، و آن چنین تعریف می‌شود:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} \quad (16.8)$$

^{۲۴}audio CDs

^{۲۵}continuous-time Fourier transform

^{۲۶}discrete-time Fourier transform (DTFT)

^{۲۷}discrete Fourier transform (DFT)

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi nk/N} \quad (17.8)$$

که در آن N تعداد نقاط بکار رفته در نمونه‌برداری $X(\phi)$ است. ما از حرف k برای نمایش ضرایب DFT بدون دیمانسیون در حوزه "فرکانس گسسته"، استفاده می‌کنیم (این به فرکانس‌های فیزیکی از صفر تا f_s در فواصل $\frac{1}{N-1} f_s$ مربوط است).

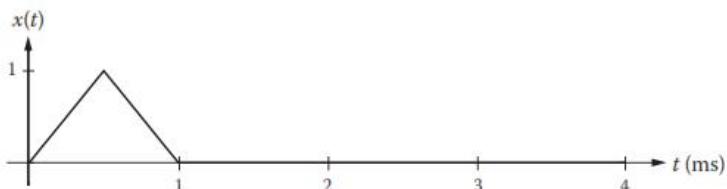
توجه کنید که هر کجا DTFT بازه $\frac{1}{2} - \frac{1}{2} \leq \phi \leq k \leq N-1$ بکار برد شود، نسبت به محور y ها متقارن است، و طبق قرار داد DFT در بازه $0 \leq k \leq N-1$ و بنابراین برای $k \geq 0$ یعنی سمت راست صفحه، تعریف می‌شود. دلیل آن این است که شکل موج گسسته را فقط وقتی N فرد باشد، می‌توان حول محور y متقارن در نظر گرفت. ضمناً DFT اغلب با استفاده از N زوج (معمولًاً توانهای دو) محاسبه می‌نمایند. دلیل آن است که DFT حاصل حول $k = \frac{N-1}{2}$ بجای $k = 0$ متقارن است.

روابط قبلی ممکن است پیچیده و انتزاعی بنظر برسد، اما مزایای بیشتر استفاده از DTFT بجای DFT چنین است: DFT را می‌توان با استفاده از الگوریتم تبدیل فوریه سریع ^{۲۸} FFT محاسبه کرد [۱۶.۸]. تبدیل فوریه سریع FFT از نظر محاسباتی راه کارآمدی برای محاسبه رابطه (۱۶.۸) با استفاده از فقط عملیات جمع و ضرب عددی است، بنابراین محاسبه حوزه فرکانس را با پردازشگر سیگنال دیژیتالی DFP ارزان قیمت فراهم می‌کند. با وجودی که اصطلاح FFT اغلب با قابل تعویض است، در حقیقت FFT یکی از چندین راه محاسباتی است که می‌توان برای محاسبه DFT بکار برد.

متلب تابع $X = fft(x)$ دارد که n نقطه از بردار x که در آن n تعداد عناصر در X است را بر می‌گرداند. بر عکس، می‌توانید بطور ضمنی تعداد نقاط بDTFT را در FFT با اضافه کردن آرگومان دوم $fft(X, 64)$ مشخص کنید. بنابراین، تعداد 64 نقطه FFT را روی بردار X محاسبه می‌کند؛ اگر X از 64 عنصر داشته باشد، در آن صورت عناصر اضافی در محاسبات FFT دور انداخته می‌شود. اگر X کمتر از 64 عناصر داشته باشد، در آن صورت X بطور خودکار با اضافه کردن صفر محاسبه FFT را انجام میدهد. متلب ضمناً تابع $ifft$ را برای محاسبه عکس DFT فراهم کرده است.

مثال ۳.۸

فرض کنید $x(t)$ یک شکل موج مثلثی مانند شکل (۱۶.۸) با عرض یک میلی ثانیه و ارتفاع یک ولت و کل طول شکل موج چهار میلی ثانیه است. ضمناً فرض کنید سرعت نمونه‌برداری ۵ کیلو نمونه بر ثانیه است.



شکل ۳.۸: شکل موج زمان پیوسته مثلثی $x(t)$

۱. تابع $|X(\phi)|$ (Mقدار DTFT) را مستقیماً از تعریف رابطه (۱۶.۸) محاسبه کنید.
۲. تابع $|X(k)|$ (Mقدار DFT) را مستقیماً از تعریف رابطه (۱۶.۸) محاسبه کنید.
۳. تابع $|X(k)|$ را با استفاده از تابع $fft()$ متلب محاسبه کنید.

^{۲۸}fast Fourier transform (FFT)

```
% Example_8_3.m
% Calculate the DTFT and DFT for a triangular waveform.
% First, define x(t)
triangle_width = 1e-3;
triangle_height = 1;
T = 1e-4;
triangle_slope = triangle_height/(triangle_width/2);
fs = 1/T;
t = 0:T:4e-3;
n = t/T;
N = length(n);
% define discrete triangle waveform x[n]
x = zeros(1,N);
x(1:6) = 0:triangle_slope*T:triangle_height;
x(6:11) = 1:-triangle_slope*T:0;
% define phi for DTFT with arbitrary granularity of 200
% steps
phi=-.5:.005:.5;
% define k for DFT as same length as x[n]
k = 0:N-1;
% initialize DTFT and DFT vars
X_DTFT = zeros(1,length(phi));
X_DFT = zeros(1,length(k));
% calculate DTFT and DFT based on definitions (sum of
% complex exponentials)
for i=1:length(n)
    X_DTFT = X_DTFT + x(i)*exp(-j*2*pi*n(i)*phi);
    X_DFT = X_DFT + x(i)*exp(-j*2*pi*n(i)*k/N);
end
% Calculate 41-point FFT with MATLAB's fft() function
X_FFT = fft(x);
% plot results
subplot(4,1,1);
stem(n,x);
xlabel('n'); ylabel('x[n]');
subplot(4,1,2);
plot(phi,abs(X_DTFT));
xlabel('\phi'); ylabel('X(\phi) (DTFT)');
subplot(4,1,3);
stem(k,abs(X_DFT));
xlabel('k'); ylabel('X[k] (DFT)');
subplot(4,1,4);
% plot FFT horiz axis as proportion of sampling rate
stem(k/(N-1) * fs,abs(X_FFT));
xlabel('f = fs*(k/(N-1)) (Hz)'); ylabel('X[k] (FFT)');

```

- نتیجه آن در شکل (۹.۸) نشان داده شده است. برخی ملاحظات بقرار زیر است:
- نوع زمان گسسته تابع مثلثی را با $x[n]$ نمایش داده و در شکل (۹.۸)(الف) ترسیم شده است.
 - تابع DTFT سیگنال $x[n]$ در شکل (۹.۸)(ب) نشان داده شده است و دارای شکل $\frac{\sin^2 x}{x^2}$ میباشد. این دارای مفهوم است، زیرا شکل موج مثلثی را میتوان با کانولوشن دو پالس مستطیلی بدست آورده، و بنابراین در حوزه فرکانس مانند ضرب دو تابع $\frac{\sin x}{x}$ خواهد بود.
 - نتایج برای DTFT شکل (۹.۸)(ب) و DFT شکل (۹.۸)(ج)، یک شکل و شبیه هم هستند، اما DFT فرکانس‌های "منفی" را سمت راست بجای سمت چپ رسم کرده است. البته، چون DFT همواره متقارن است، بسادگی میتوانیم نصف سمت راست منحنی را حذف یا چشم پوشی کنیم.
 - نتایج جمع نمائی مختلط DFT شکل (۹.۸)(ج) و متلب محاسبه شده FFT شکل (۹.۸)(د)، بطوری که انتظار میرود، یکسان هستند. البته، در شکل (۹.۸)(د)، محور x را بر حسب $\frac{k}{N-1}f_s$ (بر حسب هرتز) بجای پارامتر بدو دیمانسیون k ، مدرج کردیم. از این‌رو برای شکل محتوای بهتری را همراه دارد.

۸.۸ یک فیلتر زمان گسسته ساده

بطوری که در بخش ۳.۸ تشریح شد، یک فیلتر سیستمی است که رابطه ورودی و خروجی نوعی تابع وابسته به فرکانس است. در طراحی فیلتر زمان پیوسته، یک مدار آنالوگ که شامل خازنهای، سیم‌پیچها، و تقویت‌کننده‌ها است نوعاً برای ساخت پاسخ مناسب بکار می‌رود. فیلترها با پاسخ فرکانسی‌شان $H(f)$ و پاسخ ضربه‌ای مربوطه $h(t)$ مشخص می‌شوند، مشروط بر آنکه رابطه ورودی و خروجی بوسیله $Y(f) = H(f)X(f)$ در حوزه فرکانس یا $y(t) = h(t)*x(t)$ در حوزه زمان تعیین شود. (در اینجا نماد $*$ نمایشگر کانولوشن است). اگر $x(t)$ برابر تابع دلتا دیراک $\delta(t)$ باشد، در آن صورت $y(t) = h(t)$ است زیرا هر تابع که با $\delta(t)$ کانولوشن شود خودش می‌شود. در فیلترهای زمان گسسته، (t) و $h(t)$ رای راحتی ریاضی است زیرا امکان تولید تابع دلتا در آزمایشگاه (که معمولاً با یک پالس باریک تقریب زده می‌شود) امکان ندارد. بعبارت دیگر، در سیستم‌های زمان گسسته، نشان دادن تابع دلتا $\delta[n]$ دقیقاً امکان‌پذیر است (که معمولاً آنرا تابع واحد نمونه^{۲۹} یا دلتای کرونکر^{۳۰} می‌نامند). تابع دلتای گسسته $\delta[n]$ یک رشتہ است که در $n=0$ مقدارش واحد و بقیه جاهای دیگر صفر و بنابراین قابل اجرا است. (به شکل ۱۰.۸ مراجعه کنید). بعلاوه، هر رشتہ گسسته دیگری مانند $x[n]$ را می‌توان بر حسب جمع دلتاهای انتقال یافته بیان کرد. برای مثال، $[n]x[n]$ که در شکل (۱۱.۸)(الف) نشان داده است را می‌توان از نظر ریاضی چنین بیان کرد:

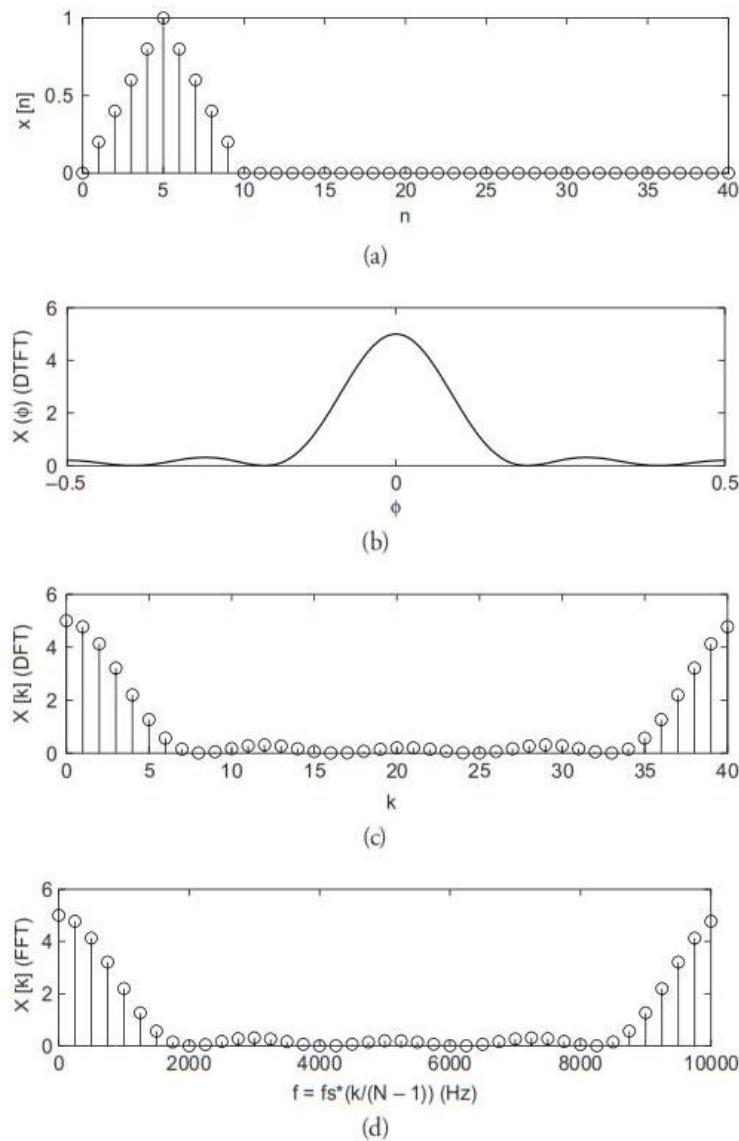
$$x[n] = 0.5\delta[n] + 1.5\delta[n-1] + \delta[n-2] - 0.5\delta[n-3]$$

بنابراین، اگر بتوانیم یک سیستم خطی با رفتاری معلوم برای $\delta[n]$ با پاسخ ضربه‌ای مربوطه $[h[n]$ طرح کنیم، در آن صورت میتوانیم سیستم را برای پردازش $x[n]$ بکار ببریم. در این حالت، ما دقیقاً خروجی را میدانیم زیرا $x[n]$ جمع وزن شده توابع دلتا است. این مفهوم با سیستم شکل (۱۱.۸)(ب) روش‌گشته که اجرای پاسخ ضربه‌ای شکل (۱۱.۸)(الف) است. برای درک این موضوع، این سیستم از تقویت‌کننده‌ها، جمع‌کننده‌ها و عناصر تاخیردهنده^{۳۱} که هدفش ذخیره سازی مقدار سیگنال دقیقاً در لحظه n است، (آنرا با نماد "D" در دیاگرام نشان داده‌ایم) ساخته می‌شود. در عمل، از ریجیسترها کامپیوتر برای عناصر تاخیردهنده، و ضرب و جمع محاسبات کامپیوتري استفاده می‌کنیم. سیستم چنین کار می‌کند:

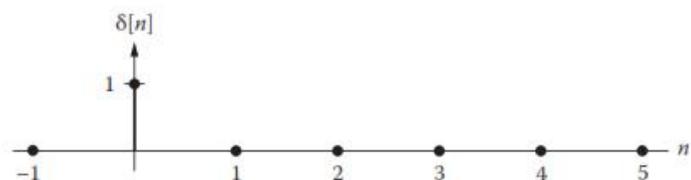
^{۲۹}unit sample function

^{۳۰}Kronecker delta function

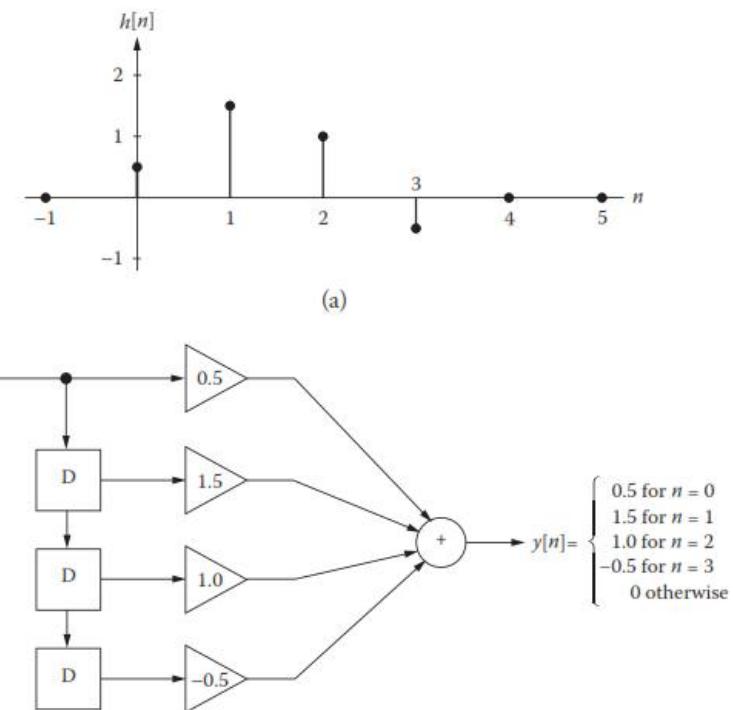
^{۳۱}delay elements



شکل ۹.۸: (الف) نوع زمان گسسته شکل موج مثلثی شکل (۸.۸). (ب) مقدار DTFT تابع $|X(\phi)|$. (ج) مقدار تابع $|X[k]|$. (د) محاسبه شده بوسیله تابع $fft()$ متلب. توجه کنید که در حالت آخر، محور فرکانس از ۰ تا f_s بجای N مقیاس بندی شده است.



شکل ۱۰.۸: نوع زمان گسسته تابع دلتا شامل یک نمونه به ارتفاع واحد در $n = 0$ و صفر هر جای دیگری است.



شکل ۱۱.۸: (الف) نمونه‌ای از پاسخ ضربه‌ای $h[n]$ برای یک فیلتر زمان گسسته. (ب) مدار فیلتر با پاسخ ضربه‌ای $h[n]$ که شامل سه عنصر تاخیر دهنده، چهار ضرب کننده، و یک جمع کننده است.

۱. ابتداء، فرض کنید که تمام عناصر تاخیر دهنده مقدار صفر دارند، در این صورت، خروجی صفر است.

۲. در "زمان" $n = 0$ ، ورودی تابع دلتا یک است؛ بنابراین، جمع خروجی سیستم برابر ۰.۵ است.

۳. در $n = 1$ ، ورودی به صفر برمی‌گردد. البته، یک از $n = 0$ در اولین عنصر تاخیر دهنده ذخیره می‌شود؛ بنابراین، جمع خروجی برابر ۱.۵ است.

۴. در $n = 2$ ، ورودی صفر می‌ماند، و یک به عنصر تاخیری بعدی انتقال داده شده، و باعث می‌شود که خروجی برابر ۱.۰ شود.

۵. در $n = 3$ ، یک به عنصر تاخیری آخری انتقال داده شده، و جمع خروجی برابر ۰.۵ می‌شود.

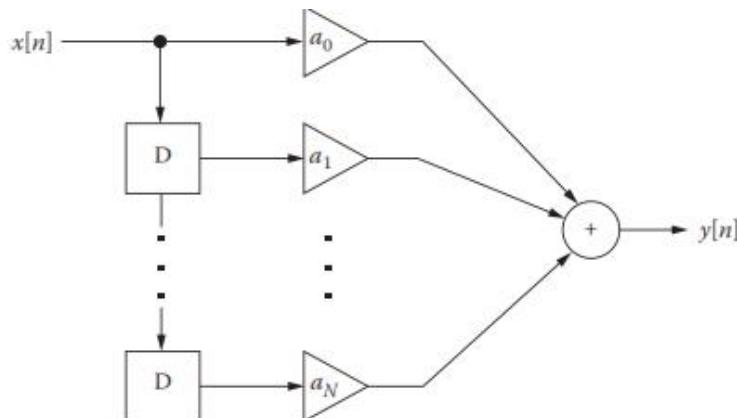
۶. سرانجام، برای $n > 3$ ، تمام عناصر تاخیری دوباره صفر شده و خروجی به صفر برمی‌گردد.

بنابراین، رشتہ خروجی این سیستم نسبت به n بر اثر تابع دلتای ورودی $\{0.5, 1.5, 1.0, -0.5\}$ خواهد بود، که مربوط به رشتہ $h[n]$ است که در شکل (۱۱.۸)(الف) میباشد.

این نوع فیلتر در شکل (۱۲.۸) برای N عنصر تاخیر دهنده که آنرا مرتبه N آم (یا معادل $N + 1$ سر^{۳۲}) پاسخ ضربه‌ای محدود FIR مینامند عمومیت داده شده است. از نظر ریاضی، این سیستم کانولوشن $y[n] = h[n] * x[n]$ را محاسبه می‌کند.

^{۳۲}tap

^{۳۳}finite impulse response (FIR)

شکل ۱۲.۸: مدار کلی ساخت یک فیلتر FIR با $N + 1$ سر.

چندین روش برای طراحی فیلتر FIR [۴] وجود دارد، و ساده‌ترین روش در بخش ۵.۸ تشریح شده است. ابتداء با دیوار آجری پاسخ فرکانسی ایده‌آل $H_I(f)$ شروع می‌کنیم. تبدیل $H_I(f)$ به حوزه زمان پاسخ ضربه‌ای غیرعلی (۱۳.۷) است که با اعمالتابع پنجره دنباله $h_I(t)$ پاسخ ضربه‌ای قابل تحقق $(h_R(t))$ بدست می‌اید. ضمناً $h_R(t)$ را انتقال داده تا در $t = 0$ شروع شود. بنابراین، برای یک فیلتر ایده‌آل $h_I(t) = \frac{\sin 2\pi f_0 t}{\pi t}$ و نوع قابل تحقق پاسخ ضربه‌ای مربوط به آن خواهد بود

$$h_R(t) = \begin{cases} \frac{\sin 2\pi f_0(t - \frac{t_w}{2})}{\pi(t - \frac{t_w}{2})} & \text{برای } t \leq t_w \\ 0 & \text{در غیر این صورت} \end{cases} \quad (18.8)$$

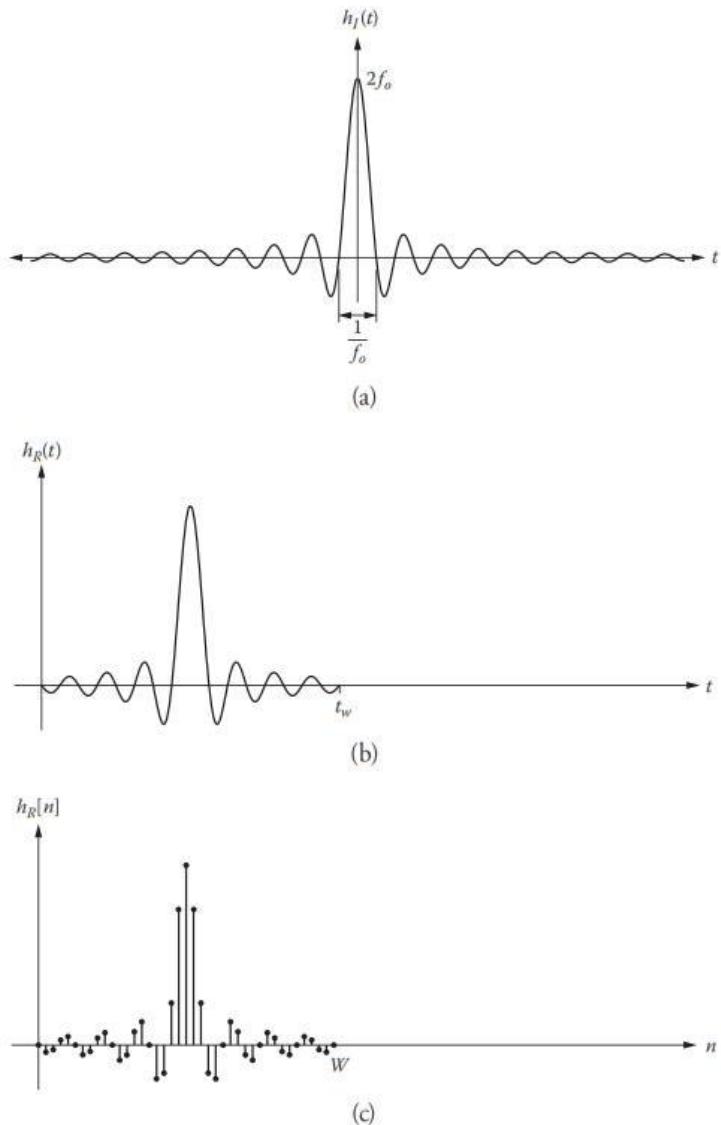
که در آن t_w عرض پنجره مرخم مستطیلی است (به شکل ۱۳.۸-ب مراجعه کنید) برای ساخت یک فیلتر زمان گسسته با سرعت نمونه‌برداری f_s (و فواصل زمانی نمونه‌برداری $\frac{1}{f_s}$)، ما $(T = \frac{1}{f_s})$ را با اعمال $t = nT$ به $h_R[n]$ تبدیل می‌کنیم. بنابراین،

$$h_R[n] = \begin{cases} \frac{\sin 2\pi \phi_0(n - \frac{W}{2})}{\pi(n - \frac{W}{2})} & \text{برای } n \leq W \\ 0 & \text{در غیر این صورت} \end{cases}$$

که در آن $\phi_0 = f_0/f_s = t_w/T$ فرکانس قطع فیلتر زمان گسسته و $W = t_w/T$ عرض پنجره است (به شکل ۱۳.۸-ج مراجعه کنید). توجه کنید که تعداد سرها در فیلتر FIR برابر $W + 1$ مطابق با W سر برای $n = 1, 2, 3, \dots, W$ بعلاوه یکی بیشتر در $n = 0$ است.

مثال ۴.۸:

یک فیلتر FIR پنجره مستطیلی، زمان گسسته پائین‌گذر، طرح کنید که فرکانس قطع آن $f_0 = 1000 \text{ Hz}$ باشد. سرعت نمونه‌برداری را $f_s = 40000$ نمونه در ثانیه و تعداد سرها را ۴۰۱ فرض کنید. فیلتر ۴۰۱ سر بدین معنی است که پاسخ ضربه‌ای باید بطول $W = 400T$ باشد. بنابراین، برای پنجره مستطیلی، ما نوع قابل تحقق فیلتر را با انتقال $h_I(t)$ باندازه $200T$ و مرخم و بریدن دم‌های آن برای $t < 0$ و $t > 400T$ بددست آید، انجام میدهیم. فیلتر قابل تحقق ۴۰۱ نقطه $h_R[n]$ برابر



شکل ۱۳.۸: (الف) پاسخ ضربه‌ای ایدهآل $h_I(t)$ برای یک فیلتر پائین‌گذر ایدهآل. (ب) با انتقال و مرخص کردن پاسخ ضربه‌ای قابل تحقق $h_R(t)$ را بدست می‌آوریم. (ج) با نمونه‌برداری کردن از $h_R(t)$ پاسخ ضربه‌ای $h_R[n]$ را بدست می‌آوریم که میتوان آنرا بصورت فیلتر FIR زمان گسسته ساخت.

است با:

$$h_R[n] = \begin{cases} \frac{\sin 2\pi\phi_0(n - \frac{W}{2})}{\pi(n - \frac{W}{2})} & \text{برای } n \leq W \\ 0 & \text{در غیر این صورت} \end{cases}$$

که در آن $\phi_0 = f_0/f_s$ فرکانس قطع فیلتر زمان گستته، و $W = t_w/T$ عرض پنجره است (به شکل (ج) مراجعه کنید). توجه کنید که تعداد سرهای فیلتر FIR برابر $W + 1$ است. که W سر برای $n = 1, 2, 3, \dots, W$ و بعلاوه یک، آن هم برای $n = 0$ است.

```
% Example_8_4.m
% This example implements an FIR low-pass filter using a
% truncated sinc function (equivalent to an ideal filter
% with a rectangular window)
clc; clear;
% define sampling rate:
fs = 40000; T=1/fs;
% define cutoff frequency of low-pass filter:
fo = 1000;
% define frequencies of input signal x(t):
input_freqs = [ 500 1500 3000 ];
% define filter order (N):
taps = 401; N = taps-1;
% Note: from this point forward, we will operate in the
% dimensionless discrete-time "n" domain. First, we
% redefine the filter cutoff frequency in terms of the
% dimensionless Phi parameter of the DTFT:
phi_o = fo / fs;
% Define hr[n] as a shifted sinc function over 401 taps
for i=1:taps
    n = i-1;
    shifted_n = ( n -N/2 );
    hr(i) = sin(2 * pi * phi_o * shifted_n) ...
        / (pi * shifted_n);
    % Note: since sin(x)/x is singular for x=0, catch
    % this case and insert the appropriate limit value.
    % Otherwise, our hr[n] won't work because it will
    % have a single "NaN" value at the peak.
    if shifted_n==0
        hr(i) = 2*phi_o;
    end
end
% Run the filter for a long time so that we can be sure
% to reach the steady state. Arbitrarily choose 20000
% samples.
max_n = fs/2;
n = 0:max_n;
```

```
% Define x[n]. First, rescale the input frequencies to be
% fractions of fs:
input_phis = input_freqs / fs;
x = zeros(1,length(n));
for i=1:length(input_phis)
x = x + sin(2*pi*input_phis(i)*n);
end
% Shift x[n] to "turn on" at a time greater than the
% number of taps of the filter. This is equivalent to
% setting the initial conditions of the filter to zero.
x_shifted = zeros(1,length(n));
x_shifted(taps:length(n)) = x(1:length(n)-N);
% Compute y[n] by simulating the delay circuit of Figure
% 8.12. This is equivalent to convolving x[n] with hr[n].
y = zeros(1,length(n));
for i=taps:length(n)
for j=1:taps
% calculate the value for each tap and add to the
% running total
y(i) = y(i) + hr(j)*x_shifted(i-j+1);
end
end
% Convert the discrete signals hr[n],x[n],and y[n] to
% the time domain by rescaling the horizontal axis and
% connecting the dots.
interesting_interval = 1400*T;
subplot(3,1,1);
plot(T*n(1:taps),hr);
% rescale the axes to see the interesting parts:
axis([0 interesting_interval -.02 .06]);
ylabel('h_R(t)');
title('FIR filter w/ rectangular window,time domain');
subplot(3,1,2);
plot(T*n,x_shifted);
axis([0 interesting_interval -3 3]);
ylabel('x(t)');
subplot(3,1,3);
plot(T*n,y);
ylabel('y(t)'); xlabel('t (sec)');
axis([0 interesting_interval -1.5 1.5]);
figure;
% See how good our filter is by computing a 4096-point
% DFT of x[n] and y[n] and plotting on semilog axes.
% Rescale the horizontal axis to be in units of frequency
% instead of the dimensionless "k" parameter of the DFT.
```

```

fft_points = 4096;
scaled_freqs = 0:fs/(fft_points-1):fs;
interesting_freq = 6000;
subplot(3,1,1);
HR = fft(hr,fft_points);
semilogy(scaled_freqs,abs(HR));
ylabel('|H_R(f)|');
% rescale the axes to see the interesting parts:
axis([0 interesting_freq 1e-4 1e1]);
title('FIR filter w/ rectangular window,freq domain');
subplot(3,1,2);
% Calculate the fft on the end part of x[n] and y[n] so
% that we are sure that we are in the steady state
% portion.
x_end = x(end-fft_points+1:end);
X=fft(x_end);
semilogy(scaled_freqs,abs(X));
axis([0 interesting_freq 1e-1 1e4 ]);
ylabel('|X(f)|');
subplot(3,1,3);
y_end = y(end-fft_points+1:end);
Y=fft(y_end);
semilogy(scaled_freqs,abs(Y));
axis([0 interesting_freq 1e-1 1e4 ]);
ylabel('|Y(f)|');
xlabel('f (Hz)');
%-----

```

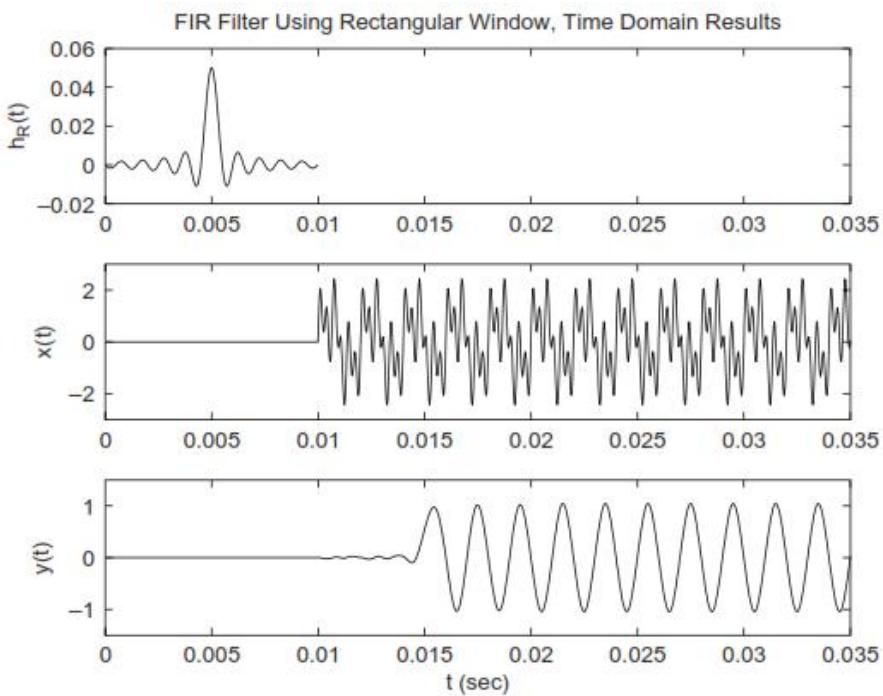
سیگنال‌های حوزه زمان $x(t)$ و $y(t)$ برای این فیلتر در شکل (۱۴.۸) نشان داده شده است. توجه کنید که این نمودارهای تبدیل سیگنال‌های حوزه زمان گسسته به حوزه زمان پیوسته با تغییر مقیاس nT در روی محور افقی انجام می‌شود. علاوه، با وجودی که ما ما 0.5 sec را برای $y(t)$ محسوبه کردیم، ما فقط بخش مطلوب (برای $t < 35 \text{ ms}$) را رسم نمودیم. بطوری که انتظار می‌رود، سیگنال چند فرکانسی $x(t)$ توسط فیلتر پائین‌گذر فرکانس‌های زیر 500 Hz باقی مانده است و این در نمودار $y(t)$ مشهود است.

در شکل (۱۵.۸)، نتیجه حوزه فرکانس برای فیلتر با اجرای تابع fft روی $x[n]$ و $y[n]$ نشان داده‌ایم. در این حالت، ۴۰۹۶ نقطه FFT روی این سیگنالها اجرا کرده‌ایم. در این صورت، این نتایج فرکانس گسسته در حوزه فرکانس مقیاس‌بندی شده و روی محور نیمه لگاریتمی برای ناحیه مطلوب ($f < 6000 \text{ Hz}$) رسم شده است. به نکات زیر توجه کنید:

■ وقتی در حوزه فرکانس نگاه می‌کنیم، پاسخ ضربه‌ای مرخم ما باعث می‌شود که تضاریسی در باند توقف و عبور بوجود آید و باند توقف به صفر نرسد.

■ طیف مقداری سیگنال ورودی $X(f)$ شامل سه مولفه مشخص $500, 1500$ ، و 3000 Hz است.

■ طیف مقداری سیگنال خروجی $Y(f)$ شامل مولفه باند عبور 500 هرتز علاوه مولفه‌های تضعیف شده (اما صفر نیستند) 1500 و 3000 Hz است.



شکل ۱۴.۸: نمودار تولید شده متلب برای یک فیلتر FIR و سیگنالهای ورودی و خروجی. با اعمال $h_R(t)$ بر سیگنال خروجی $y(t)$ بدست می‌آید، که مولفه‌های بیش از 1000 Hz حذف شده است.

این فیلتر FIR، بستگی به کاربرد آن، ممکن است نتایج مطلوب و یا نامطلوبی داشته باشد. برای مثال، در کاربردهای صوتی، تضاریس باند عبور، ممکن است باعث بی دقتی صوتی در نتایج خروجی سیگنال فیلتر شده کند. عبارت دیگر، در سایر کاربردها، کمترین جهش ناحیه‌ای بین باند عبور و باند توقف (که آنرا **جهش باند می‌نامند**) یکی از خواسته‌ها با وجود تضاریس می‌باشد. یکی از روش‌های تغییر پاسخ فرکانسی فیلتر، اعمال تابع پنجره به پاسخ فیلتر ایده‌آل است. در شکل (۱۶.۸) مقایسه پنجره‌های، پاسخ ضربه‌ای ایده‌آل $h_I[n]$ ، سه پنجره مختلف $w[n]$ ، پاسخ ضربه‌ای قابل تحقق، و تبدیل فوریه را نشان میدهد. سه مثال نشان داده شده، پنجره مستطیلی (که آنرا "وگن‌باری" نیز می‌نامند) که اکنون بحث شد؛ پنجره بارلت^{۳۴} (مثالی)؛ و پنجره هامینگ^{۳۵} که یک شکل کسینوسی با مولفه DC است.

سه پنجره از نظر ریاضی بصورت زیر بیان می‌شوند:

$$w_R[n] = 1$$

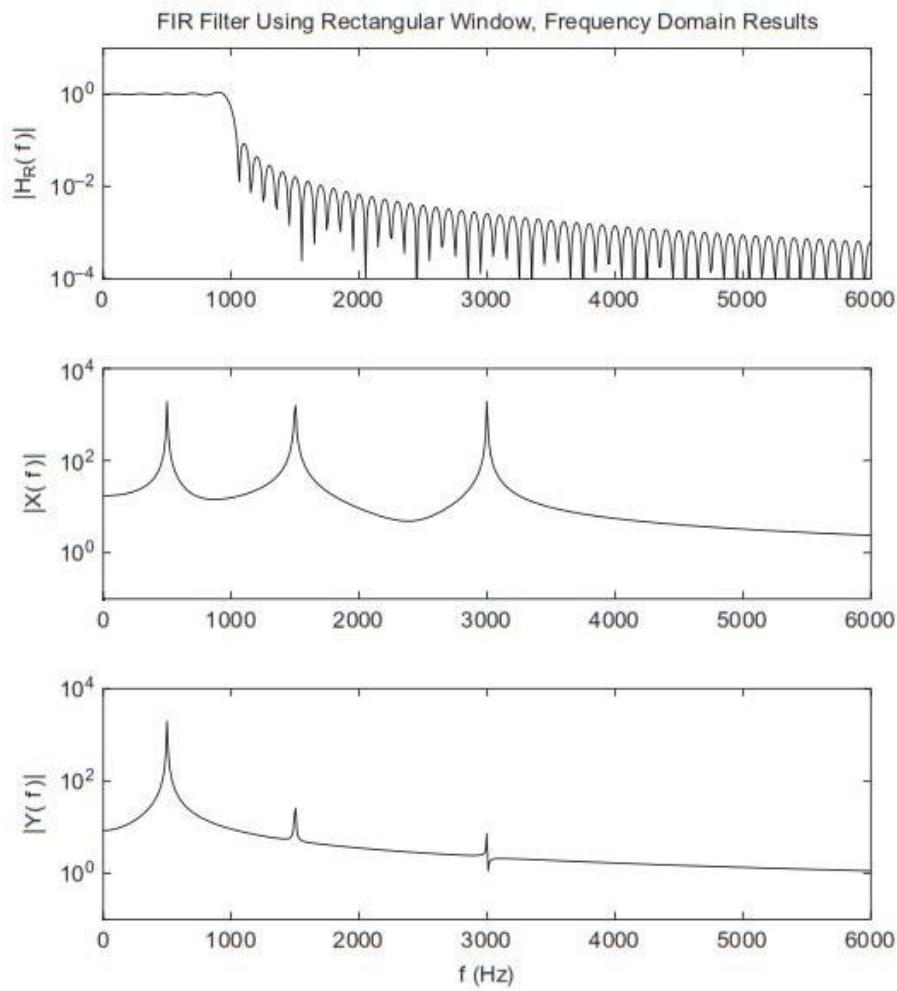
$$w_B[n] = \frac{2}{N}, \quad \frac{2}{N} - \left| n - \frac{2}{N} \right| \quad \text{بارلت}$$

$$w_H[n] = 0.56 - 0.46 \cos\left(\frac{2\pi n}{N}\right) \quad \text{هامینگ}$$

که در آن $n = 0, 1, 2, \dots, N$ است. پاسخ فرکانسی حاصله هر سه پائین‌گذر بوده، اما شکل‌های مختلفی دارند. پنجره مستطیلی تضاریس زیادی در باند عبور و باند توقف دارد لیکن تیزترین جهش باند را دارد. پنجره بارلت تضاریس کمتری نسبت به پنجره مستطیلی دارد اما جهش باند عریض‌تری است. پنجره هامینگ بیشترین اختلاف بین باند عبور و توقف را دارد؛ یعنی، بزرگترین افت بین باند عبور و اولین کلبرگ باند توقف را دارد.

^{۳۴}Bartlett

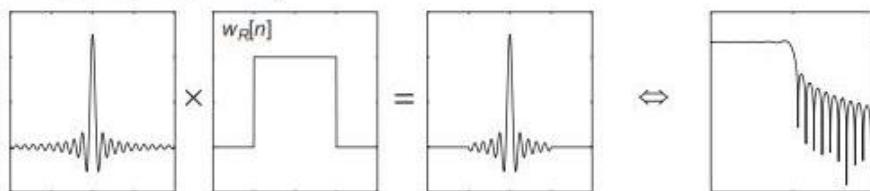
^{۳۵}Hamming



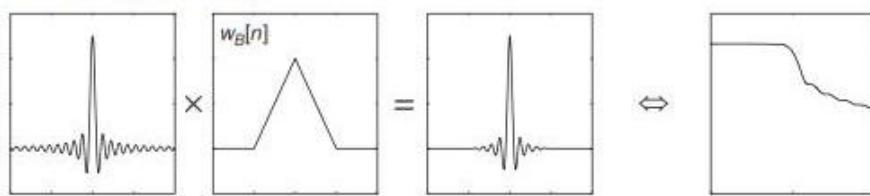
شکل ۱۵.۸: تبدیل فوریه سیگنالهای شکل ۱۴.۸ که در مقیاس نیمه‌لگاریتمی رسم شده است. نمودار بالائی پاسخ فرکانسی انتقال و مرخم شده پاسخ ضربه‌ای را نشان میدهد که دارای تضاریسی در باند عبور و توقف دارد. دو برجستگی قابل مشاهده در $|X(f)|$ (نمودار وسطی) در $|Y(f)|$ کاهش یافته است (نمودار پائینی).

$$\begin{array}{c} \text{ideal} \\ \text{impulse} \\ \text{response} \\ h_i[n] \end{array} \times \begin{array}{c} \text{window} \\ w[n] \end{array} = \begin{array}{c} \text{realizable} \\ \text{impulse} \\ \text{response} \\ h_R[n] = h_i[n]w[n] \end{array} \Leftrightarrow \begin{array}{c} \text{frequency} \\ \text{response} \\ H_R(\phi) \end{array}$$

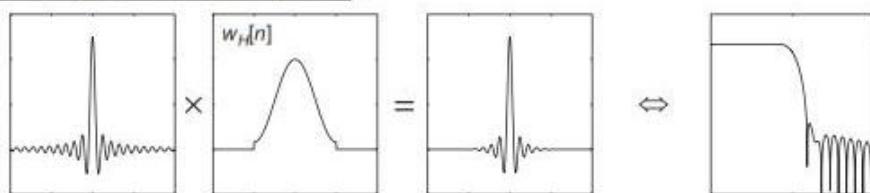
Rectangular (boxcar) window:



Bartlett (triangular) window:



Hamming (raised cosine + DC) window:

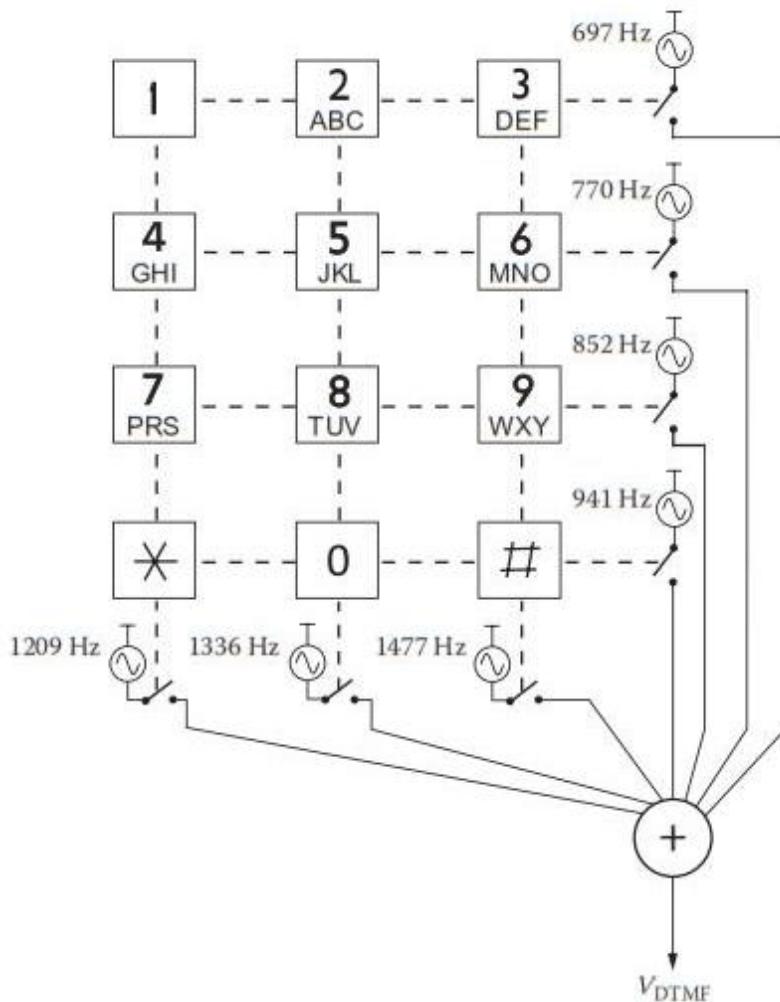


شکل ۱۶.۸: نمونه‌ای از پنجره $w[n]$ که ممکن است بر پاسخ ضربه‌ای فیلتر پائین‌گذر ایده‌آل $h_i[n]$ اعمال شده تا پاسخ ضربه‌ای قابل تحقق $h_R[n]$ برای یک فیلتر FIR بدست آید. توجه کنید که برای این که نمودار بهتر قابل خواندن باشد، شکل موجهای زمان گسسته بصورت پیوسته "با بهم پیوستن سرها" رسم شده است. پاسخ فرکانسی قابل تحقق DTFT نیز نشان داده شده است.

پروژه‌ها

۱.۸ پروژه

در سیستم تلفنی، ^{۳۶} DTMF برای کد کردن و ارسال کلیدها از گوشی تلفن به گیرنده دور روی شبکه تلفنی استفاده می‌شود. شکل (۱۷.۸) یک شمای ساده از صفحه کلید شامل هفت اتصال



شکل ۱۷.۸: صفحه کلید تلفن با دوازده دکمه، هفت اتصال، هفت مولد موج سینوسی، و جمع کننده.

و هفت منبع سینوزوئیال نشان میدهد. این اتصالها طوری ترتیب داده شده‌اند که با فشار دادن هر کلید، دو اتصال ستونی و سطحی مربوطه بسته شد، بطوری که دو منبع ولتاژ را به جمع کننده متصل می‌کند. ولتاژ خروجی حاصل V_{DTMF} شامل مجموع دو فرکانس مجزا بوده که بسوی گیرنده دور ارسال می‌شوند. برای مثال، وقتی دکمه "۲" فشار داده می‌شود، سیگنالهای با

^{۳۶} DTMF (dual-tone multifrequency)

فرکانس Hz و $697 Hz$ ۱۳۳۶ جمع شده بسوی گیرنده دور فرستاده میشوند. در گیرنده از روی فرکانسهای ارسالی تعیین میشود که در فرستنده چه دکمه‌ای تحریک شده است.

۱. تابعی بزبان متلب بنام $dtmf.m$ نوشته بطوری که برای اعداد صفر تا نه فرکانس‌های DTMF را تولید کند. (اکنون در باره دکمه‌های با علامت * و # کاری نداشته باشید). تابع شما باید یکی از آرگومان‌های ۰ تا ۹ را گرفته و یک بردار با طول مناسب شامل شکل موج تولید کند. این پارامترها را در نظر گیرید:

- سرعت نمونه‌برداری: 8000 نمونه در ثانیه
- زمان طی صوت sec : 0.25
- دامنه بیشینه سیگنال: یک ولت.

۲. از تابع DTMF برای تولید شکل موج شماره‌گیر تلفن خودتان استفاده کنید. لازم است که تابع خود را چندین بار اجرا کرده (برای هر عدد در روی صفحه کلید تلفن) و نتایج را بصورت یک بردار بهم متصل کنید.

۳. شکل موج را در یک فایل با فرمان متلب $wavwrite$ ذخیره کنید. فرمان (help wavwrite) را میتوانید بچندین روش اجرا کنید (برای اطلاعات بیشتر help wavwrite را اجرا نمائید)، اما در ساده‌ترین حالت دو آرگومان میگیرد: یکی بردار (با فرض 8000 نمونه در ثانیه) و یکی هم نام فایل. مطمئن شوید که میتوانید فایل WAV حاصله را در بلندگوی کامپیوتر خود اجرا کنید.

۴. شکل موجی که شما تولید کردید رشته مطابق DTMF تولید نکرده‌اید [۱۰۱]، زیرا بین دو صوت (تون) مدت زمان سکوت مناسب را بکار نبرده‌اید. تابع $dtmf.m$ خود را اصلاح کرده و $0.25 sec$ سکوت بین دو صوت قرار دهد. سعی کنید صوت نتیجه را روی تلفن خود (با برداشتن گوشی در روی بلندگوی کامپیوتر) آزمایش کنید و مشاهد کنید که آیا میتوانید صوت تولید شده را برای شماره‌گیری تلفن خود بکار ببرید. (توجه کنید که این فقط برای شماره‌گیرهای تون دار کار میکند نه برای تلفن همراه).

۲.۸ پروژه

سیستمی طرح کنید که سیگنالهای ارسالی آشکار نماید. فرض کنید که $f_s = 8000$ نمونه بر ثانیه است. بنحو زیر اقدام کنید:

۱. ده سیگنال آزمایشی $x_0(t)$ تا $x_9(t)$ مطابق با 10 عدد روی دکمه‌های DTMF شکل (۱۷.۸) تولید کنید. هر سیگنال باید $0.25 sec$ طول بکشد و شامل دو موج سینوسی (هر کدام با دامنه یک) مطابق با فرکانسهای سطر و ستون هر دکمه باشد. الحال تمام سیگنالهای آزمایشی شما با یکدیگر برای تعیین یک سیگنال بطول $0.25 sec$ شامل فشار تمام ده دکمه شبیه سازی بترتیب اجرا است. (بر عکس، اگر پروژه ۱.۸ را اجرا کردید، از سیگنالهایی که تولید کردید در این مسئله استفاده کنید).

۲. برنامه‌ای بزبان متلب نوشته که 1024 نقطه FFT را روی مجموعه متوالی 1024 عنصر در روی سیگنال ورودی شما انجام داده و وجود هر هفت فرکانس DTMF را آشکار نماید. نتایج آشکارسازی خود را روی صفحه نمایشگر برای عملیات FFT نشان دهید.

۳. وقتی دو کلید همزمان فشار داده میشود برای FFT چه اتفاق می‌افتد؟ اضافه کردن $0.25 sec$ سکوت بین هر فشار دکمه برای اجتناب از این مسئله است و برنامه آشکارسازی را بر میگرداند که ببینید خطای وارد حذف شده است.

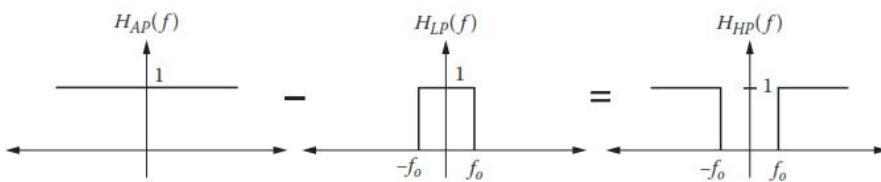
۴. اندازه پنجره FFT خود را به 512 , 256 , 128 , 64 , و 32 تقلیل دهید. کمترین پنجره‌ای که میتوانید بکار ببرید و هنوز آشکارسازی مطمئن برای تمام هفت فرکانس DTMF داشته باشید کدام است؟

^{۱۷}Tone duration

۳.۸ پروژه

یک فیلتر FIR بالاگذر با مشخصات زیر طرح کنید:

- سرعت نمونهبرداری $f_s = 8000$ نمونه در ثانیه.
- فرکانس $f_0 = 200 \text{ Hz}$
- سرها: $Taps = 201$
- پنجره: بارلت (مثلثی)



شکل ۱۸.۸: پاسخ فرکانسی فیلتر بالاگذر $H_{HP}(f)$ را میتوان با کم کردن پاسخ پائین‌گذر $H_{LP}(f)$ از پاسخ تمام‌گذر $H_{AP}(f)$ بدست آورد.

توجه کنید که فیلتر بالاگذر را میتوان (به شکل (۱۸.۸) مراجعه کنید) از اختلاف پاسخ تمام‌گذر و فیلتر پائین‌گذر، فیلتر تمام‌گذر برای تمام فرکانسها مقدار واحد دارد.

۱. پاسخ ضربه یک فیلتر تمام‌گذر را با محاسبه DTFT معکوس بر طبق رابطه (۱۵.۸) برای مقادیر $n = -2, -1, 0, 1, 2$ بدست آورید. آیا جواب شما معنی دارد؟
۲. پاسخ ضربه‌ای ایده‌آل $h[n]$ را برای فیلتر بالاگذر با کم کردن از پاسخ ضربه‌ای پائین‌گذر از تمام‌گذر محاسبه کنید.
۳. پاسخ ضربه‌ای قابل تحقق یک فیلتر بالاگذر $h_R[n]$ را از ضرب پاسخ ضربه‌ای ایده‌آل در یک پنجره مثلثی که عرض آن تعداد سرهای است و ارتفاع آن واحد است تولید کنید.
۴. سگنال ورودی آزمایشی $x(t)$ که شامل امواج سینوسی بفرکانس‌های ۶۰، ۳۰۰، و 1000 Hz است را تولید کنید. این $x(t)$ را به فیلتر بالاگذر خود اعمال کرده و خروجی $y(t)$ را توسط فیلتر FIR شکل (۱۲.۸) شبیه‌سازی کنید.
۵. تبدیل فوریه سریع FFT شامل ۱۰۲۴ نقطه را روی $(t, h_R(t))$ و $(t, y(t))$ انجام داده و طیف مقداری را محور نیمه‌لگاریتمی (محور افقی خطی و محور عمودی لگاریتمی) ترسیم کنید. چه مولفه‌های فرکانسی از درون فیلتر بالاگذر عبور می‌کنند؟

۴.۸ پروژه

یک فیلتر میان‌گذر FIR طراحی کنید که کلیدهای تلفن از سطر دوم شکل (۱۷.۸) صفحه کلید DTMF را آشکار کند. فرض کنید که $f_s = 8000$ نمونه در ثانیه، و فیلتر با ۴۰۱ سر است. بصورت زیر اقدام کنید:

۱. ابتداء، پاسخ فیلتر میان‌گذر ایده‌آل را در حوزه فرکانس تولید کرده بنحوی که مرکز باند عبور روی 770 Hz و فرکانس قطع پائینی f_L و بالائی f_H باشد. فرکانس‌های f_L و f_H از روی حذف فرکانس‌های مجاور DTMF یعنی 697 و 852 Hz تعیین می‌شود.
۲. چون تبدیل فوریه خطی است، پاسخ میان‌گذر را میتوان با کم کردن دو پاسخ پائین‌گذر تولید کرد. فیلتر میان‌گذر خود را با توجه به فرکانس قطع f_H و f_L طراحی کنید. با کم کردن پاسخ ضربه‌ای فیلترهای پائین‌گذر ایده‌آل، پاسخ ضربه‌ای فیلتر میان‌گذر ایده‌آل را بدست آورید.

۳. پاسخ ضربه‌ای قابل تحقیق این فیلتر را با ضرب پاسخ ضربه‌ای ایده‌آل در پنجره‌های شکل (۱۶.۸) تولید کنید. توضیح دهید که کدام پنجره را انتخاب می‌کنید و چرا برای کار شما بهینه است.

۴. دو سیگنال آزمایشی برای فیلتر خود مطابق با دکمه‌های شماره دو و پنج روی صفحه کلید تولید کرده و آنرا از درون فیلتر خود عبور داده و مدار فیلتر FIR شکل (۱۲.۸) را شیوه سازی کنید. نتایج حوزه زمان را رسم کرده و صحت خروجی فیلتر خود را در آشکارسازی عدد ۵ است و نه ۲ تائید کنید.

۵.۸ پروژه

با استفاده از تکنیک مثال ۲.۸، ضرایب سری فوریه نمائی را برای پالس مثلثی شکل (۸.۸) بدست آورید.

۶.۸ پروژه

مثال ۴.۸ را برای فیلتر FIR پائین‌کننده دوباره اجرا کنید. تغییرات زیر را برای پارامترهای فیلتر بکار برد و روی اثرات نمودارهای خروجی از نظر کیفی نظر خود را اعلام نمائید.

۱. سرعت نمونه برداری f_s را از ۴۰,۰۰۰ به ۱۰۰,۰۰۰ نمونه در ثانیه تغییر دهید.
۲. تعداد سرهای فیلتر را به ۱۰۱ عدد تغییر دهید.
۳. تعداد سرهای فیلتر را به ۱۶۰۱ عدد تغییر دهید.
۴. سه فرکانس ورودی را به ۲۰۰، ۵۰۰، و ۳۹,۲۰۰ Hz تغییر دهید.

فصل ۹

برازش منحنی

۱.۹ مقدمه

خیلی وقت‌ها در مهندسی لازم است که در یک آزمایش رفتار پدیده خاصی را تعیین کنیم. آزمایش ممکن است مجموعه‌ای نقاط داده تولید کند که نمایشگر رابطه بین متغیرهای پدیده باشد. ضمناً ممکن است بخواهیم این رابطه را بطور تحلیلی بیان کنیم. بیان ریاضی که توصیف کننده داده باشد را **تابع تقریبی^۱** می‌نامند. دو رویکرد برای تعیین یک تابع تقریبی وجود دارد:

۱. نمودار بصورت منحنی تابع تقریبی بیان می‌شود. در این حالت، منحنی رسم شده بطور کلی تمام نقاط داده را در بر نمی‌گیرد، اما سعی ما بر این است که خطای حاصله کمترین شود تا بهترین منحنی خورانیده شود. نمودار داده روى مختصات خطی، نیمه‌لگاریتمی، و لگاریتم-لگاریتمی را می‌توان اغلب یک شکل تقریبی برای تابع تقریبی پیشنهاد کرد.
۲. تابع تقریبی از تمام نقاط داده‌ها می‌گذرد (بطوری که در بخش ۳.۹ تشریح خواهد شد). البته، اگر برخی پراکنده‌گی در نقاط داده باشد، این تابع تقریبی ممکن است قانع کننده نباشد.

۲.۹ روش کمترین مربعات

۱.۲.۹ بهترین برازش خط مستقیم

در روش کمترین مربعات، ما بدنبال پیدا کردن بهترین برازش خط مستقیم برای مجموعه n نقاط داده $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ هستیم. میخواهیم منحنی تقریبی y_c را بصورت خط مستقیم زیر نمایش دهیم

$$y_c = c_1 + c_2 x \quad (1.9)$$

^۱approximating function

که در آن c_1 و c_2 ثابت‌های مجهول هستند که باید تعیین شوند. فرض کنید D مجموع مربعات خطای بین خط تقریبی و نقاط واقعی باشد. در این صورت،

$$D = \sum_{i=1}^n [y_i - y_c(x_i)]^2 \quad (۲.۹)$$

$$= \sum_{i=1}^n [y_i - (c_1 + c_2 x_i)]^2 \quad (۳.۹)$$

$$= [y_1 - (c_1 + c_2 x_1)]^2 + [y_2 - (c_1 + c_2 x_2)]^2 + \cdots + [y_n - (c_1 + c_2 x_n)]^2 \quad (۴.۹)$$

برای تعیین تابع تقریبی بهترین برازش خط مستقیم، کمینه کردن (مینیمم نمودن) D توسط $\frac{\partial D}{\partial c_2} = 0$ و $\frac{\partial D}{\partial c_1} = 0$ است. با گرفتن مشتق جزئی از رابطه (۳.۹) نسبت به c_1 خواهیم داشت

$$\frac{\partial D}{\partial c_1} = 0 = \sum_{i=1}^n 2[y_i - (c_1 + c_2 x_i)][-1]$$

$$0 = \sum_{i=1}^n y_i - c_2 \sum_{i=1}^n x_i - nc_1$$

یا

$$nc_1 + \left(\sum_{i=1}^n x_i \right) c_2 = \sum_{i=1}^n y_i \quad (۵.۹)$$

با گرفتن مشتق جزئی از رابطه (۳.۹) نسبت به c_2 خواهیم داشت

$$\frac{\partial D}{\partial c_2} = 0 = \sum_{i=1}^n 2[y_i - (c_1 + c_2 x_i)][-x_i]$$

$$0 = \sum_{i=1}^n x_i y_i - c_1 \sum_{i=1}^n x_i - c_2 \sum_{i=1}^n x_i^2$$

یا

$$\left(\sum_{i=1}^n x_i \right) c_1 + \left(\sum_{i=1}^n x_i^2 \right) c_2 = \sum_{i=1}^n x_i y_i \quad (۶.۹)$$

معادلات (۵.۹) و (۶.۹) یک دستگاه دو معادله دو مجهولی را بیان میدارد که میتوان آنرا با روش دترمینان (قاعده کرامر)^۲ حل کرد:

$$c_1 = \frac{\begin{vmatrix} \sum y_i & \sum x_i \\ \sum x_i y_i & \sum x_i^2 \end{vmatrix}}{\begin{vmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{vmatrix}} = \frac{(\sum y_i)(\sum x_i^2) - (\sum x_i)(\sum x_i y_i)}{n \sum x_i^2 - (\sum x_i)(\sum x_i)} \quad (۷.۹)$$

$$c_2 = \frac{\begin{vmatrix} n & \sum y_i \\ \sum x_i & \sum x_i y_i \end{vmatrix}}{\begin{vmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{vmatrix}} = \frac{n \sum x_i y_i - (\sum x_i)(\sum y_i)}{n \sum x_i^2 - (\sum x_i)(\sum x_i)} \quad (۸.۹)$$

^۲Cramer's rule

۲.۲.۹ بهترین برازش چندجمله‌ای درجه m

میتوانیم این رویکرد را برای برازش چندجمله‌ای درجه m عمومیت دهیم. در این حالت، منحنی تقریبی y_c را بصورت زیر درنظر می‌گیریم

$$y_c = c_1 + c_2x + c_3x^2 + c_4x^3 + \cdots + x_{m+1}x^m \quad (9.9)$$

که در آن $n - 1 \leq m \leq n$ تعداد نقاط داده‌ها است.

مقدار اندازه‌گیری شده‌ها (x_i, y_i) برای $i = 1, 2, \dots, n$ هستند.

فرض کنید $y_c(x_i) = y_{c,i}$ مقدار تقریبی در نقطه (x_i, y_i) باشد. در این صورت،

$$D = \sum_{i=1}^n [y_i - y_{c,i}]^2 = \sum_{i=1}^n [y_i - (c_1 + c_2x + c_3x^2 + c_4x^3 + \cdots + x_{m+1}x^m)]^2$$

برای کمینه کردن D از مشتق استفاده می‌کنیم

$$\frac{\partial D}{\partial c_1} = 0, \frac{\partial D}{\partial c_2} = 0, \dots, \frac{\partial D}{\partial c_{m+1}} = 0$$

در این صورت،

$$\frac{\partial D}{\partial c_1} = 0 = \sum_{i=1}^n 2 [y_i - (c_1 + c_2x + c_3x^2 + c_4x^3 + \cdots + x_{m+1}x^m)] [-1]$$

$$\frac{\partial D}{\partial c_2} = 0 = \sum_{i=1}^n 2 [y_i - (c_1 + c_2x + c_3x^2 + c_4x^3 + \cdots + x_{m+1}x^m)] [-x_i]$$

$$\frac{\partial D}{\partial c_3} = 0 = \sum_{i=1}^n 2 [y_i - (c_1 + c_2x + c_3x^2 + c_4x^3 + \cdots + x_{m+1}x^m)] [-x_i^2]$$

⋮

$$\frac{\partial D}{\partial c_{m+1}} = 0 = \sum_{i=1}^n 2 [y_i - (c_1 + c_2x + c_3x^2 + c_4x^3 + \cdots + x_{m+1}x^m)] [-x_i^m]$$

این مجموعه معادلات بصورت زیر خلاصه می‌شود

$$nc_1 + (\sum x_i) c_2 + (\sum x_i^2) c_3 + \cdots + (\sum x_i^m) c_{m+1} = \sum y_i$$

$$(\sum x_i) c_1 + (\sum x_i^2) c_2 + (\sum x_i^3) c_3 + \cdots + (\sum x_i^{m+1}) c_{m+1} = \sum x_i y_i \quad (10.9)$$

⋮

$$(\sum x_i^m) c_1 + (\sum x_i^{m+1}) c_2 + (\sum x_i^{m+2}) c_3 + \cdots + (\sum x_i^{2m}) c_{m+1} = \sum x_i^m y_i$$

معادله (۱۰.۹) را می‌توان از طریق روش حذف گوسی (که در فصل سوم بحث شد) حل کرد.

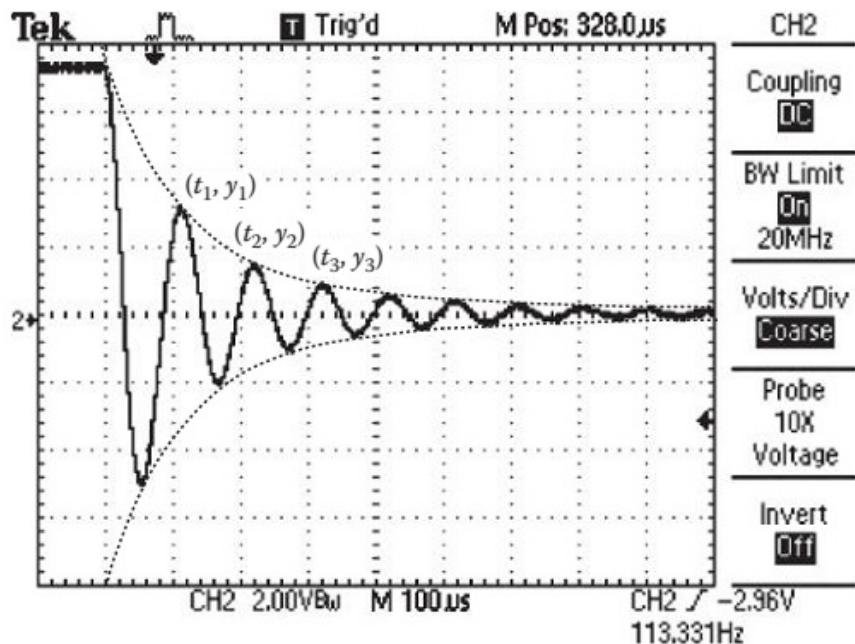
متلب از خطای مربع میانگین ^۳ (MSE) بعنوان اندازه‌گیری دقیق دقت که با آن چند جمله‌ای برای توصیف داده‌ها خورانیده (برازش) می‌شود، استفاده می‌کند. خطای مربع میانگین (MSE) چنین تعریف می‌شود:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_{c,i})^2 \quad (11.9)$$

که در آن $y_{c,i} = y_c(x_i)$ است.

برای تعیین بهترین درجه چندجمله‌ای، برنامه خود را بازه m های مختلف اجرا کرده و از آنها که کمترین (MSE) را دارند را استفاده کنید.

^۳mean square error (MSE)



شکل ۱.۹: عکس صفحه اسیلوسکوپ پاسخ پله واحد ولتاژ خازن یک مدار (RLC).

۳.۹ برازش منحنی با توابع نمائی

خیلی از دستگاهها (مثل، قطعات نیمه‌هادی) را می‌توان با توابع نمائی مدل کرد. اگر داده‌های آزمایشی شما در این مقوله باشد، می‌توان با تابع زیر برازش نمود

$$y_c = \alpha_1 e^{-\alpha_2 x} \quad (12.9)$$

که در آن α_1 و α_2 مقادیر ثابت حقیقی هستند.
بیایید فرض کنیم که مجموعه n نقطه داده $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ وجود دارد. در این صورت، $z_i = \ln y_i$ و $c_1 = \ln \alpha_1 - \alpha_2 x_i$ و $c_2 = -\alpha_2$ را خواهیم داشت. بنابراین، رابطه (۱۲.۹) نسبت به z_c خطی می‌شود، یعنی،

$$z_c = c_1 + c_2 x \quad (13.9)$$

برای نقاط داده $(x_1, z_1), (x_2, z_2), \dots, (x_n, z_n)$ ، مجموعه جدید نقاط $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ را حواهد شد.

بطوری که در بخش قبل بدست آوردیم، بهترین برازش تقریبی منحنی خط مستقیم با روش کمترین مربعات میدهد

$$c_1 = \frac{(\sum z_i)(\sum x_i^2) - (\sum x_i)(\sum x_i z_i)}{n \sum x_i^2 - (\sum x_i)^2} \quad (14.9)$$

$$c_2 = \frac{n \sum x_i z_i - (\sum x_i)(\sum z_i)}{n \sum x_i^2 - (\sum x_i)^2} \quad (15.9)$$

در این صورت، $c_1 = \exp(c_1)$ و $c_2 = -c_2$ است. روش کمترین مربعات (MSE) را می‌توان با معادله (۱۱.۹) تعیین کرد.

این تحلیل را می‌توان برای تعیین ضریب میرائی ξ در یک مدار (RLC) موازی (در پیوست آ تحلیل شده است) بکار برد. این را می‌توان با آزمایش پاسخ پله واحد خروجی اسیلوسکوپ (شکل ۱.۹) انجام

$$y = y_0 e^{-\xi \omega_n t} \quad (16.9)$$

که در آن

- ولتاژ خازن y =

- ولتاژ اولیه خازن y_0 = (بستگی به شرایط اولیه دارد).

- ضریب میرائی ξ =

- فرکانس طبیعی $\omega_n = \frac{1}{\sqrt{LC}}$ =

مقایسه رابطه (۱۵.۹) با (۱۰.۹)، در می‌یابیم که

$$\alpha_1 = y_0$$

$$\alpha_2 = \xi \omega_n$$

با قرار دادن t بجای x . بنابراین

$$\xi = \frac{\alpha_2}{\omega_n} = \alpha_2 \sqrt{LC} \quad (17.9)$$

با اندازه‌گیری n نقطه داده روی پوش، یعنی $(t_1, y_1), (t_2, y_2), \dots, (t_n, y_n)$ ، میتوانیم بهترین مقدار برازش را برای α_2 برای بهترین تخمین ضریب میرائی ξ تعیین کنیم.

۴.۹ تابع polyfit متلب

با توجه به برازش منحنی با یک چند جمله‌ای متلب اصطلاح "رکرسیون چند جمله‌ای"^۴ را عنوان میکند. تابع polyfit بردار $m+1$ ضریب را که نمایشگر بهترین برازش چند جمله‌ای درجه m برای مجموعه نقاط داده (x_i, y_i) است، را بما میدهد. مرتبه ضرایب مربوط به توان نزولی x است، یعنی

$$y_c = a_1 x^m + a_2 x^{m-1} + a_3 x^{m-2} + \dots + a_m x + a_{m+1} \quad (18.9)$$

بردار $A = \text{polyfit}(X, Y, m)$ شامل ضرایبی است که باید با اجرای $A = (a_1, a_2, \dots, a_m, a_{m+1})$ بدست آید. در اینجا $X = (x_1, x_2, \dots, x_n)$ و $Y = (y_1, y_2, \dots, y_n)$ هستند.

متلب تابع polyval(a, x) را برای محاسبه مقادیر تقریبی $y_{c,i}$ در (x_1, x_2, \dots, x_n) برای خورانیدن ضرایب $(a_1, a_2, \dots, a_{m+1})$ تدارک دیده است:

$$y_{c,i} = a_1 x_i^m + a_2 x_i^{m-1} + a_3 x_i^{m-2} + \dots + a_m x_i + a_{m+1}$$

برای محاسبه MSE، بطوری که در مثال ۱.۹ مشاهده می‌شود، تابع polyval مفید خواهد بود.

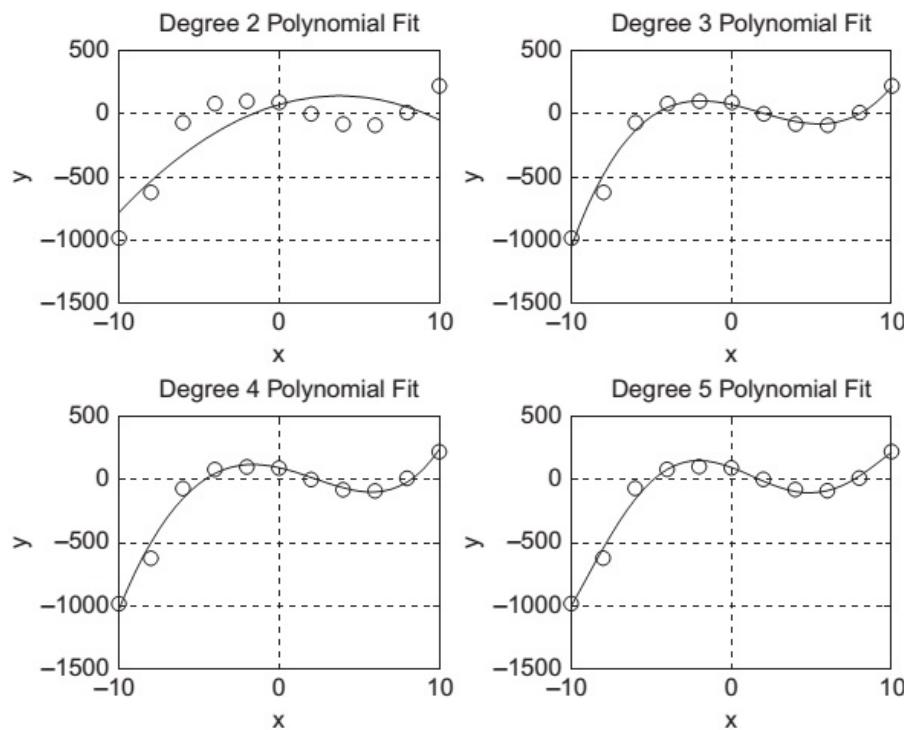
مثال ۱.۹:

نقاط داده زیر داده شده است. بهترین برازش چند جمله‌ای درجه دوم، سوم، چهارم، و پنجم را بدست آورید. ضمناً خطای مربع میانگین را محاسبه کنید.

$$(x, y) = (-10, -980), (-8, -620), (-6, -70), (-4, 80), (-2, 100) \\ (0, 90), (2, 0), (4, -80), (6, -90), (8, 10), (10, 220)$$

^۴polynomial regression

```
% Example_9_1.m
% This program determines the best fit polynomial curve
% passing through a given set of data points for
% polynomial degrees 2 thru 5.
clear; clc;
% Provided data set
X = [ -10 -8 -6 -4 -2 0 2 4 6 8 10 ];
Y = [-980 -620 -70 80 100 90 0 -80 -90 10 220 ];
% Fit to polynomials of degree m:
m=2:5;
% Create a second set of x points over the same interval
% in order to plot our fitted curve
X2 = -10:0.5:10;
% Vector to keep track of MSE (to print at end)
MSE=zeros(length(m));
for i=1:length(m)
    fprintf('m = %i \n',m(i));
    % an mth-degree polynomial has m+1 coefficients:
    A = zeros(m(i)+1);
    % do the fit
    A = polyfit(X,Y,m(i));
    % compute the MSE
    Yc = polyval(A,X);
    MSE(i) = (1/length(X)) * sum((Y-Yc).^2);
    % print table
    fprintf('x y yc \n');
    fprintf('-----\n');
    for j=1:length(X)
        fprintf('%5.0f %5.0f %8.2f \n',...
            X(j),Y(j),Yc(j));
    end
    fprintf('\n\n');
    % plot
    subplot(2,2,i);
    plot(X2,polyval(A,X2),X,Y,'o');
    xlabel('x'); ylabel('y'); grid;
    axis([-10 10 -1500 500]);
    title(sprintf('Degree %d polynomial fit',m(i)));
    end
    % print computed MSE
    fprintf('m MSE \n');
    fprintf('-----\n');
    for i=1:length(m)
        fprintf('%g %10.2f \n',m(i),MSE(i))
    end
```



شکل ۹.۲: با استفاده از تابع `polyfit` برای چند جمله‌ای درجه دوم، سوم، چهارم، و پنجم.

خروجی برنامه

| m | MSE |
|---|----------|
| 2 | 32842.37 |
| 3 | 2659.97 |
| 4 | 2342.11 |
| 5 | 1502.95 |

بطوری که انتظار می‌رود، MSE با افزایش مرتبه برازش چندجمله‌ای کاهش می‌یابد. برازش چندجمله‌ای در شکل (۹.۲) ترسیم شده است.

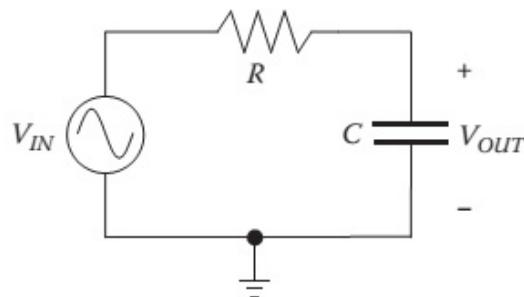
مثال ۹.۱

شکل (۹.۳) یک مدار RC و حدول (۴.۹) مجموعه دامنه‌های ولتاژ و فاز اندازه‌گیری شده حالت ماندگار در آزمایشگاه را برای V_{OUT} و V_{IN} در فرکانس‌های داده شده نشان میدهد.

میخواهیم داده را با بهترین پاسخ فرکانسی مرتبه اول ممکن برازش کنیم. مدار RC را میتوان یک فیلتر پائین‌گذر مرتبه اول با پاسخ فرکانسی زیر در نظر گرفت:

$$H(f) = \frac{n_0}{d_0 + j2\pi f} \quad (9.1)$$

که در آن $H = V_{OUT}/V_{IN}$ نسبت خروجی به ورودی نسبت به فرکانس بوده، و n_0 و d_0 ضرایب صورت و مخرج میباشند که باید بدست آورده شوند. چون رابطه (۹.۱) دارای چندجمله‌ای در مخرج



شکل ۳.۹: مدار RC با ورودی ولتاژ سینوزوئیدال V_{IN} و خروجی V_{OUT} .

جدول ۱.۹: اندازه‌گیری آزمایشگاهی پاسخ فرکانسی مدار RC

| فرکانس (Hz) | V_{IN} (ولت؛ پیک تو پیک) | V_{OUT} (ولت؛ پیک تو پیک) | اختلاف فاز بین V_{IN} و V_{OUT} |
|-------------|-------------------------------|--------------------------------|--|
| 0 | 5 | 5 | 0 |
| 2,000 | 5 | 2.2 | -60 |
| 4,000 | 5 | 1.2 | -75 |
| 6,000 | 5 | 0.8 | -80 |
| 8,000 | 5 | 0.6 | -83 |
| 10,000 | 5 | 0.5 | -85 |

است، ما فرم آنرا تغییر داده تا مطابق رابطه (۱۸.۹) شده و در آن صورت میتوانیم polyfit را اعمال کنیم.

$$\frac{1}{H} = \frac{1}{n_0} j 2\pi f + d_0/n_0 \quad (20.9)$$

تطبیق جملات بین روابط (۲۰.۹) و (۱۸.۹) نشان میدهد که $a_2 = d_0/n_0$ و $a_1 = j 2\pi/n_0$ است. بنابراین، عکس پاسخ فرکانسی $\frac{1}{H}$ را می‌توان با استفاده از (۲۰.۹) برآورده کرد. در اینجا f بردار فرکانس‌های اندازه‌گیری شده، H برداری شامل مقادیر اندازه‌گیری شده V_{OUT}/V_{IN} است. برنامه متلب چنین است:

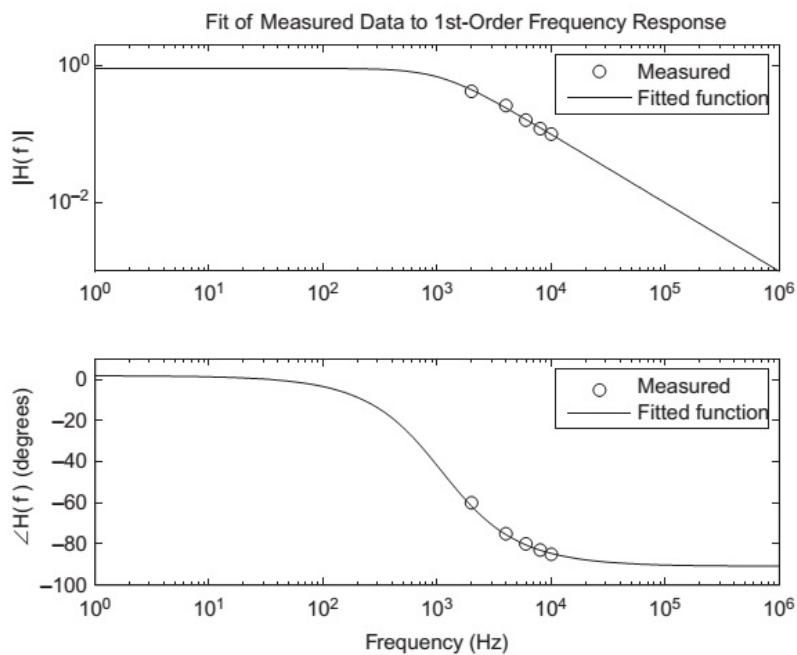
```
% Example_9_2.m
% Fit lab-measured freq response to 1st order response
clear; clc;
% Lab-measured data from Table 9.1:
f = [ 0 2e3 4e3 6e3 8e3 10e3 ];
V_in = [ 5 5 5 5 5 5 ];
V_out = [ 5 2.1 1.3 .8 .6 .5 ];
Phi_degrees = [ 0 -60 -75 -80 -83 -85 ];
% Ratio of V_out to V_in is the magnitude of the
% frequency response:
H_mag = V_out ./ V_in;
% The phase is the measured phase difference (converted
```

```
% to radians)
H_phase = Phi_degrees * pi / 180;
% Express the complex freq response H as:
% mag * exp(j*phase)
H = H_mag .* exp(j*H_phase);
% Do polynomial fit to 1/H
A=polyfit(f,1./H,1);
% Compute coefficients of fitted freq response based
% polyfit results
n0 = 2*pi*j/A(1);
d0 = 2*pi*j*A(2)/A(1);
% Print fitted frequency response:
fprintf('Fitted frequency response:\n');
fprintf('H(f) = %.0f / (j*2*pi*f + %.0f)\n',n0,d0);
fprintf('Pole location: %.0f rad/sec (%.0f Hz)\n',...
d0,d0/(2*pi));
% For comparison purposes, calculate the fitted curve
% with fine precision. Use 100 log-spaced points from 1Hz
% to 1MHz
f_fine = logspace(0,6,100);
H_fit = n0 ./ (j*2*pi*f_fine + d0);
H_fit_mag = abs(H_fit);
H_fit_phase = angle(H_fit);
% Create Bode plot:
subplot(2,1,1);
loglog(f,H_mag,'o',f_fine,H_fit_mag);
axis([ 1e0 1e6 1e-3 2]);
legend('measured','fitted function');
ylabel('|H(f)|');
title('Fit of measured data to 1st-order freq response');
subplot(2,1,2);
semilogx(f,(180/pi)*H_phase,'o',...
f_fine,(180/pi)*H_fit_phase);
axis([ 1e0 1e6 -100 10]);
legend('measured','fitted function');
ylabel('\angle H(f) (degrees)'); xlabel('frequency (Hz)');
-----
```

خروجی برنامه

Fitted frequency response:
 $H(f) = 6204/(j*2*pi*f + 6842)$
 Pole location: 6842 rad/sec (1089 Hz)

نمودار نتیجه در شکل (۴.۹) نشان داده شده است.



شکل ۴.۹: برازش پاسخ فرکانسی برای مدار مرتبه اول RC .

۵.۹ خط نقطه‌گذار مکعبی

فرض کنید مجموعه‌ای نقاط داده شده است. تمام جستجو تقریب منحنی‌های درجه چندجمله‌ای‌هایی است که نقاط تولید شده مجاز نمی‌باشند. برای مثال، فرض کنید میدانیم که خواص ویژه‌ای که با داده‌ها نمایش داده شده است (از قبیل ولتاژ یا دمای مطلق) باید مثبت باشد، و تمام جستجو منحنی تقریب چندجمله‌ای‌ها تولید برخی مقادیر منفی می‌کنند. برای این حالت،تابع تقریب چندجمله‌ای قانون کننده و مناسب نیستند. روش خط نقطه‌گذار مکعبی^۴ این مسئله را حذف می‌کند.

فرض کنید مجموعه $n+1$ نقاط داده (x_i, y_i) برای $i = 1, 2, \dots, n+1$ داده شده است. روش خط نقطه‌گذار یک مجموعه n تابع مکعبی بطوری که $y(x)$ با تابع مکعبی مختلف در هر n فاصله را نمایش دهد، و مجموعه توابع مکعب از درون $1+n$ نقطه داده بگذرد را میخواهیم داشته باشیم.

برای انجام این عمل سعی می‌کنیم شب و انحناء مطابق با هر زوج تابع مکعبی که نقاط داده را بهم متصل می‌کند باشد. توجه کنید که انحناء برابر است با:

$$K = \frac{\pm \frac{d^2y}{dx^2}}{\left[1 + \left(\frac{dy}{dx} \right)^2 \right]^{3/2}} \quad \text{انحناء} \quad (21.9)$$

مشاهده میشود که اگر ما مجموعه معادلات زیر، [معادلات (۲۲.۹)]، را ارضاء کنیم، بطور خودکار

⁴cubic splines

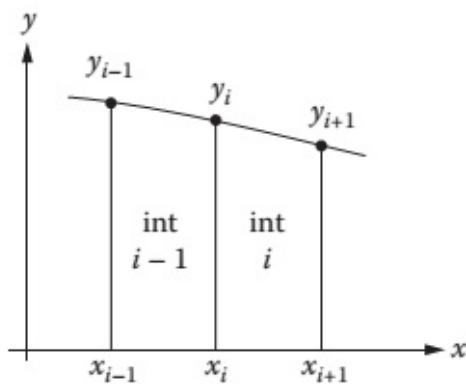
شیب و انحنای را در نقاط اتصال ارضا خواهیم کرد. بنابراین،

$$[y(x_i)]_{int\ i-1} = [y(x_i)]_{int\ i} \quad (22.9)$$

$$[y'(x_i)]_{int\ i-1} = [y(x_i)']_{int\ i} \quad (22.9)$$

$$[y''(x_i)]_{int\ i-1} = [y(x_i)'']_{int\ i}$$

در نقطه x_i مقدار y'_i و y''_i باید برای فاصله چپ ($int\ i - 1$) و فاصله راست ($int\ i$) یکسان باشد. در فاصله $x_{i-1} \leq x \leq x_i$ (به شکل ۵.۹ مراجعه کنید)، داریم



شکل ۵.۹: دو فاصله مجاور در درون یابی خط نقطه‌گذار مکعبی

$$y(x) = A_{i-1} + B_{i-1}(x - x_{i-1}) + C_{i-1}(x - x_{i-1})^2 + D_{i-1}(x - x_{i-1})^3 \quad (23.9)$$

در اینجا A_{i-1} , B_{i-1} , C_{i-1} , و D_{i-1} ضرایب چندجمله‌ای مکعبی برازش در بین x_{i-1} و x_i هستند.

در فاصله (i) ، $x_i \leq x \leq x_{i+1}$ داریم

$$y(x) = A_i + B_i(x - x_i) + C_i(x - x_i)^2 + D_i(x - x_i)^3 \quad (24.9)$$

که در آن A_i , B_i , C_i , و D_i ضرایب چندجمله‌ای مکعبی برازش در بین x_i و x_{i+1} هستند.

این $n - 1$ معادله و $n + 1$ مجهول میدهد.

بنابراین، مقادیر برای $\frac{d^2y}{dx^2}$ در x_1 و x_{n+1} را باید فرض کرد تا بتوان دو معادله آخری را تولید نمود. چندین گزینش برای انتخاب این دو معادله وجود دارد:

- الف- فرض کنید $0 = y(x_{n+1}) = y(x_1)$. این در خیلی خط نقطه‌گذرهای اجباری با رویکر خطوط مستقیم در انتهایها استفاده می‌شود.

- ب- فرض کنید $y(x_2) = y(x_n) = y(x_1)$ و $y(x_{n+1}) = y(x_1)$. این خط نقطه‌گذرهای اجباری با رویکر سهموی در نقاط انتهایی است.

۶.۹ تابع interp1 برای برازش خط نقطه‌گذار مکعبی

در متلب تابع ساخته شده داخلی interp1 برای درون یابی بین نقاط داده با روش خط نقطه‌گذار مکعبی دارد. دستور العمل آن چنین است:

$$Yi = \text{interp1}(X, Y, Xi, 'spline') \quad \text{کاربرد:}$$

که در آن X, Y مجموعه نقاط داده، و Xi مجموعه مقادیر X که آن مقادیر Yi را بما می‌دهد.

مثال ۳.۹

```
% Example_9_3.m
% This program uses interpolation by cubic splines to
% upsample an audio signal.
clear; clc;
% Define original data points for y(t) (time in microsec)
orig_t=[0 4 8 12 16 20 24 28 32 36 40 44 48 52 56];
orig_y=[.7 .9 .9 .7 .3 0 -.3 -.7 -.7 -.3 0 .3 .7 .7 .3];
% Define upsampled time points
upsampled_t = 0:60;
% Calculate interpolated data points using cubic splines
interp_y = interp1(orig_t,orig_y,upsampled_t,'spline');
% Print output to screen
fprintf('UPSAMPLING VIA CUBIC SPLINE FIT\n');
fprintf('time (microsec) orig sample upsample\n');
fprintf('-----\n');
for i=1:length(upsampled_t)
% Try to find the corresponding timepoint in the
% original dataset for this upsampled timepoint with
% MATLAB's "find" function.
j = find(orig_t == upsampled_t(i));
if j>0
% if we found original value, print both orig and
% interpolated.
fprintf('%8.2f %10.1f %10.3f\n',...
upsampled_t(i),orig_y(j),interp_y(i));
else
% otherwise, just print interpolated value
fprintf('%8.2f %10.3f \n',...
upsampled_t(i),interp_y(i));
end
end
plot(orig_t,orig_y,'o',upsampled_t,interp_y);
xlabel('t (microsec)'); ylabel('y(t)'); grid;
title('Upsampling with Cubic Spline Interpolation');
legend('original','upsampled','Location','SouthWest');
%-----
```

خروجی برنامه

```
UPSAMPLING VIA CUBIC SPLINE FIT
time(microsec)    original sample    upsample
0.00                  0.7          0.700
1.00                  0.7          0.770
2.00                  0.7          0.827
```

| | | |
|-------|-----|-------|
| 3.00 | | 0.870 |
| 4.00 | 0.9 | 0.900 |
| 5.00 | | 0.918 |
| 6.00 | | 0.923 |
| 7.00 | | 0.917 |
| 8.00 | 0.9 | 0.900 |
| 9.00 | | 0.872 |
| 10.00 | | 0.830 |
| 11.00 | | 0.774 |
| 12.00 | 0.7 | 0.700 |
| 13.00 | | 0.608 |
| 14.00 | | 0.506 |
| 15.00 | | 0.400 |
| 16.00 | 0.3 | 0.300 |
| . | | . |
| . | | . |

۷.۹ برازش منحنی با سری فوریه

فرض کنید مجموعه داده آزمایشی $u(t)$ مانند نمودار شکل (۶.۹) را تولید کرده است و میخواهیم رابطه‌ای تحلیلی نزدیک به این داده‌ها آنرا برازش کند. بیانید فرض کنیم که بازه داده‌ها در فواصل متقاضی $[-L, L]$ است. اگر اینطور نبود ما مبداء را قدری انتقال میدهیم. در این حالت، مبداء داده‌ها محور افقی برای $0 \leq t \leq 10.5 \text{ sec}$ بوده، و داده‌ها را با قرار دادن $x = t - 5.25$ انتقال میدهیم. در این صورت، حوزه x را به 70 قسمت مساوی تقسیم کرده و هر قسمت $\Delta x = 10.5/70 = 0.15 \text{ sec}$ می‌شود.

بنابراین $x_{i+1} - x_i$ در تمام حوزه یکسان است. سعی در برازش منحنی تقریبی چند جمله‌ای به این داده‌ها نمی‌تواند منحنی که شباهت زیادی با این داده‌ها باشد انجام شود. البته، استفاده از سری فوریه رابطه‌ای نسبتاً تحلیلی برای تقریب این داده‌ها بما میدهد. اگر u_c منحنی تقریبی باشد، سری فوریه خواهد بود

$$u_c(x) = a_0 + \sum_{m=1}^{\infty} \left[a_m \cos\left(\frac{m\pi x}{L}\right) + b_m \sin\left(\frac{m\pi x}{L}\right) \right] \quad (25.9)$$

که در آن

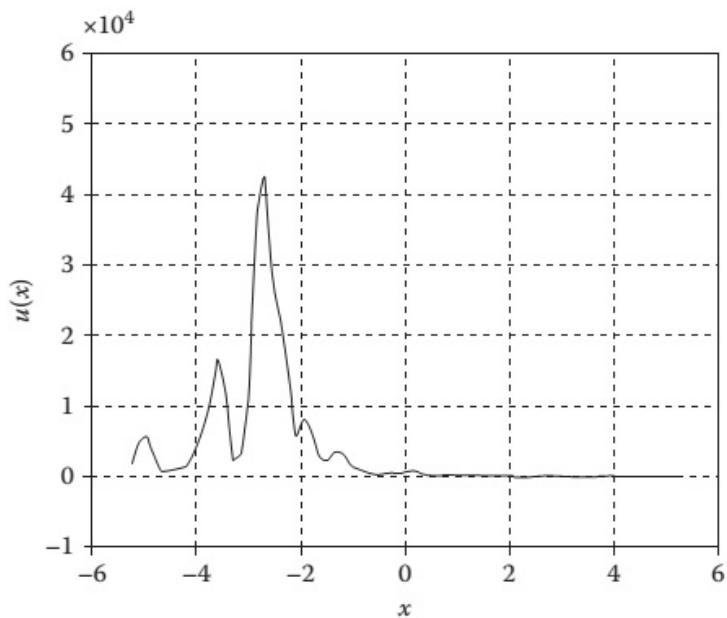
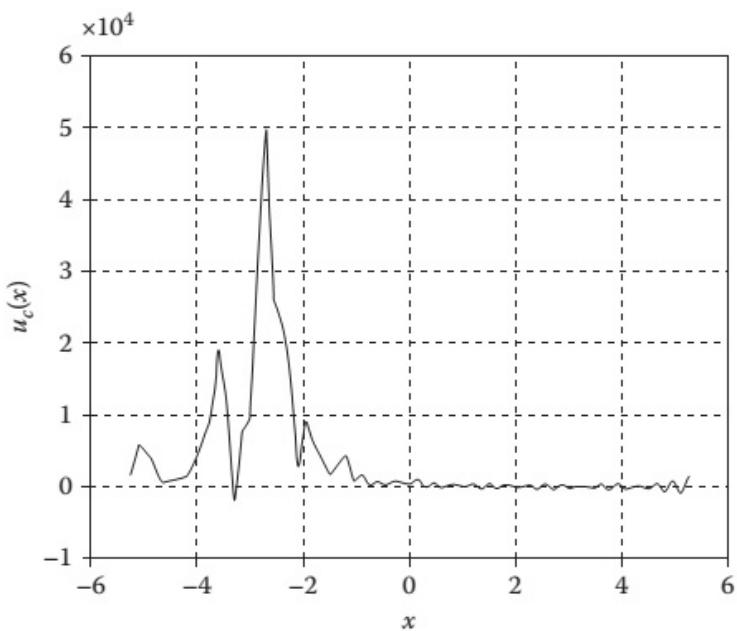
$$a_0 = \frac{1}{2L} \int_{-L}^{L} u(x) dx \quad (26.9)$$

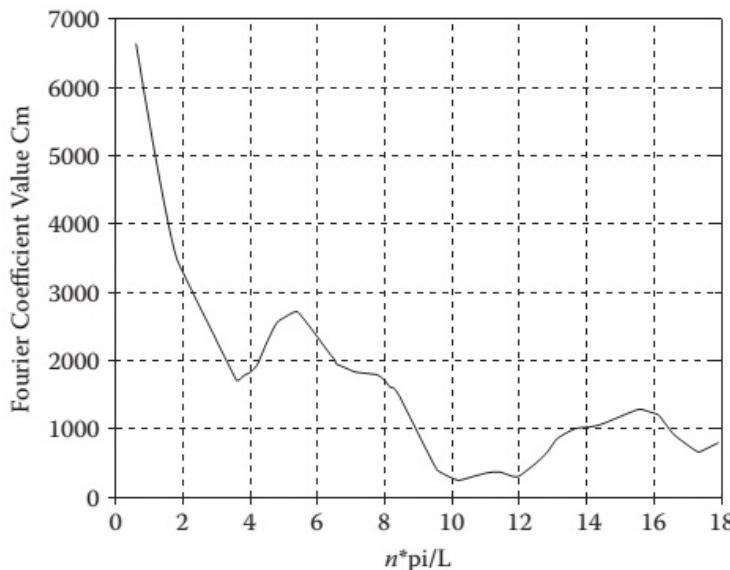
$$a_m = \frac{1}{L} \int_{-L}^{L} u(x) \cos\left(\frac{m\pi x}{L}\right) dx \quad (27.9)$$

$$b_m = \frac{1}{L} \int_{-L}^{L} u(x) \sin\left(\frac{m\pi x}{L}\right) dx \quad (28.9)$$

با استفاده از 30 جمله از این سری و روش انتگرال‌گیری سیمپسون، منحنی تقریبی شکل (۷.۹) بدست می‌آید. جملات $a_m \cos\left(\frac{m\pi x}{L}\right) + b_m \sin\left(\frac{m\pi x}{L}\right)$ را می‌توان با استفاده از اتحاد مثلثاتی $a \cos \beta + b \sin \beta = \sqrt{a^2 + b^2} \sin(\beta + \phi)$ ساده کرد، و خواهیم داشت

^۷Fourier series

شکل ۶.۹: داده آزمایشی u در مقابل x شکل ۷.۹: برآورد فوریه u_c در مقابل x برای داده شکل (۶.۹).

شکل ۸.۹: دامنهای ضرایب سری فوریه در مقابل $\frac{m\pi}{L}$

$$c_m = \sqrt{a_m^2 + b_m^2}$$

منحنی c_m بر حسب $\frac{m\pi}{L}$ (با صرفنظر کردن از عامل ϕ) در شکل (۸.۹) ترسیم شده است.

پروژه‌ها

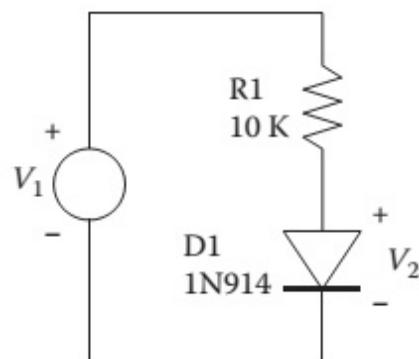
۱.۹ پروژه

شکل (۹.۹) یک مدار دیود- مقاومت را با استفاده از دیود (D1) نیمه‌هادی سیلیکن ۱N914 و یک مقاومت ده کیلو اهم (R1) نشان میدهد. جدول (۷.۹) اندازه‌گیری آزمایشگاهی V_2 را بر حسب ولتاژ ورودی اعمالی V_1 در دمای اطاق ($300 K$) نشان میدهد.

۱. با استفاده از قانون ولتاژ کیرشهف جریان دیود i_D را بر حسب V_1 و V_2 محاسبه کنید.
 ۲. با استفاده از روشی که در بخش ۳.۹ تشریح شد، برازش کمترین مربعات i_D و $v_D (= V_2)$ فرمول زیر را بدست آورید
- $$i_D = I_S \exp\left(\frac{v_D}{V_T}\right) \quad (29.9)$$
- در این حالت، از داده سطحی برای پیدا کردن بهترین برازش مقادیر I_S و V_T استفاده کنید. روی یک دستگاه مختصات منحنی برازش و داده آزمایشگاهی را رسم کنید.
۳. مقدار $V_T = kT/q$ بعنوان ولتاژ حرارتی^۴ معروف است (k : ثابت بولتزمن^۵ و q : واحد بار الکتریکی است). برای بهترین برازش خود بازاء V_T ، چه مقدار دمای T (بر حسب کلوین) است؟ آیا این مقدار معقول بنظر میرسد؟

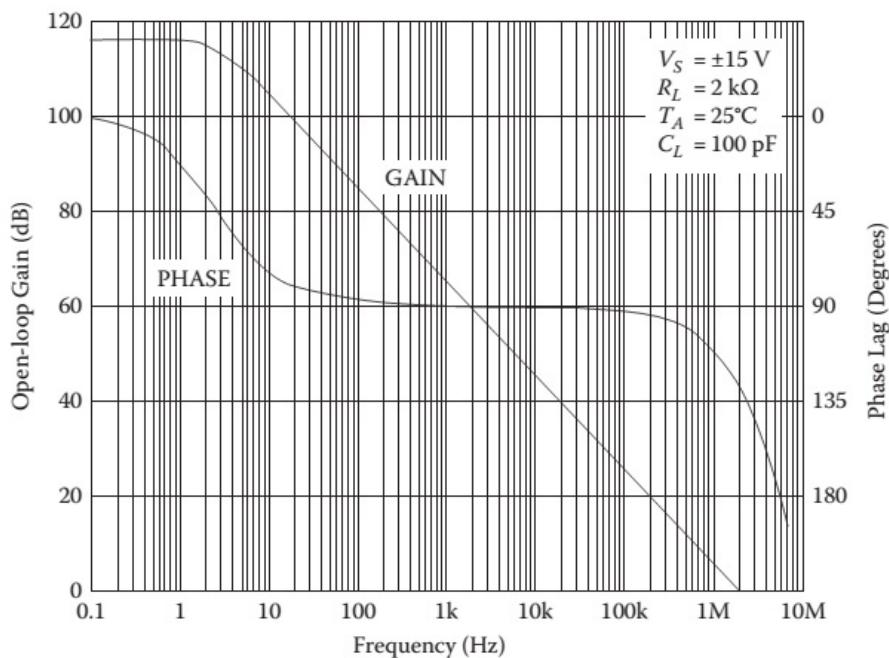
^۴thermal voltage

^۵Boltzmann constant

شکل ۹.۹: مدار دیود- مقاومت برای اندازه‌گیری منحنی $V-I$ دیود.

جدول ۹.۹: اندازه‌گیری آزمایشگاهی مدار دیود- مقاومت

| ولت V_1 | ولت V_2 |
|-----------|-----------|
| 0.189 | 0.189 |
| 0.333 | 0.317 |
| 0.393 | 0.356 |
| 0.819 | 0.464 |
| 1.067 | 0.487 |
| 1.289 | 0.501 |
| 1.656 | 0.518 |
| 1.808 | 0.522 |
| 2.442 | 0.541 |
| 3.949 | 0.566 |
| 4.971 | 0.579 |
| 6.005 | 0.595 |
| 6.933 | 0.588 |
| 7.934 | 0.602 |
| 9.014 | 0.607 |
| 10.040 | 0.613 |
| 11.009 | 0.619 |
| 15.045 | 0.634 |
| 19.865 | 0.647 |
| 24.64 | 0.657 |
| 29.79 | 0.666 |



شکل ۱۰.۹: پاسخ فرکانسی تقویت‌کننده عملیاتی 741

۴. رابطه دقیقتر برای مدل کردن دیود چنین است

$$i_D = I_S \exp\left(\frac{v_D}{nV_T}\right)$$

که در آن n ضریب ایدهآل^۹ است. مقدار بهترین برازش n برای $K = 300$ را بدست آورید.

۵. مدل $V - I$ را برای دیود بصورت زیر تکمیل کنید

$$i_D = I_S \left[\exp\left(\frac{v_D}{nV_T}\right) - 1 \right]$$

آیا منطقی است که ما از جمله ۱- در برازش منحنی صرفنظر کنیم؟ چرا؟

۲.۹ پروژه

پاسخ فرکانسی حلقه باز اندازه‌گیری تقویت‌کننده عملیاتی 741 در شکل (۱۰.۹) نشان داده شده است.

۱. اندازه (بر حسب دسی‌بل) و فاز (بر حسب درجه) پاسخ فرکانسی را مستقیماً با اندازه کمیری از روی نمودار بدست آورید. توجه کنید که خواندن فاز (سمت راست نمودار) بصورت "تاخیر فاز" است و بدین معنی است که قرائت باید بصورت درجه منفی تفسیر شود. آنرا بصورت دو بردار در متلب وارد کنید.

۲. معمولاً بهره تقویت‌کننده A را بر حسب دسی‌بل (dB) بیان می‌کنند، که در آن

$$A_{dB} = 20 \log A$$

^۹ideality factor

قرائت دسیبل خود را با تقسیم بر بیست و سی پس بتوان ده رسانیدن، آنرا به بهره مطلق تبدیل کنید. ضمناً، قرائت درجه را به رادیان نیز تبدیل نمایید. در این صورت، پاسخ فرکانسی را بصورت قطبی مختلط بیان کنید. بخارط داشته باشید که $|z|e^{j\phi} = z$ و در آن $|z|$ مقدار و ϕ زاویه فاز است.

۳. با استفاده از تابع `polyfit` مطلب، پاسخ فرکانسی را بصورت زیر برازش کنید.

$$H(f) = \frac{d_0}{(j2\pi f)^2 + n_1 j2\pi f + n_0}$$

راهنمایی: برازش خود را برای $1000 \leq f$ هرتز انجام داده و سپس برای فرکانس‌های بالاتر هر بار نقاط را اضافه کنید. چون چندجمله‌ای در مخرج کسر است، سعی کنید $1/H(f)$ را بجای $H(f)$ برازش کنید.

۴. محل قطب‌ها را برای برازش $(f) H$ با پیدا کردن ریشه‌های مخرج بدست آورید.

۵. نتایج اندازه‌گیری و نتایج برازش شده را یک دستگاه مختصات رسم کنید.

۳.۹ پروژه

برای `orig_t` و `orig_Y` از داده‌های مثال ۳.۹ استفاده کرده و سری فوريه را انجام دهيد. ابتداء را بنحوی انتقال داده که در نواحی $-L \leq t \leq L$ (در اينجا $L = 28 \mu sec$ است) متقارن شود. سپس، مراحل زير را انجام دهيد.

۱. برنامه‌اي بзиان متلب نوشته که m ضريب فوريه را محاسبه کند. از روش انتگرال‌گيري سيمپسون (در فصل پنجم شرح داده شده است) برای محاسبه ضرایب $a_0, a_1, a_2, \dots, a_m$ و b_1, b_2, \dots, b_m در روابط (۲۶.۹) - (۲۸.۹) استفاده کنيد.

۲. سه مجموعه ضرایب سري فوريه را بازء $m = 2, m = 4$ و $m = 6$ محاسبه کنيد.

۳. سه منحنی برازش بازء $m = 2, 4, 6$ با قراردن ضرایب در رابطه (۲۵.۹) تولید کنيد. هر سه را روی يك دستگاه رسم کنيد. وقتی منحنی‌های برازش شده را رسم مي‌کنيد، از ۱۵۰ نقطه روی محور افقی در محدوده $-L \leq t \leq L$ - استفاده نمائيد. (بهجای ۱۵ نقطه توسط (orig_t). ضمناً، `orig_Y` را همان دستگاه رسم کرده و از علامت `X` برای نشان دادن نقاط استفاده کنيد).

۴. برای حالت $m = 6$ ، آنچه اتفاق میافتد را از نظر کيفی توضیح دهيد.

فصل ۱۰

بهینه‌سازی

۱.۱۰ مقدمه

هدف از بهینه‌سازی، بیشینه یا کمینه کردن یکتابع مانند f (که آنرا تابع هدف^۱ می‌نامند) است. احتمالاً با پیدا کردن بیشینه یا کمینه یکتابع با استفاده از مشتق ریاضی آشنا هستید. البته، وقتی محدودیتهای^۲ و قیودی روی جوابهای موجود^۳ قرار میدهیم، مسئله خیلی غامض و پیچیده می‌شود. بعنوان مثال، یک شرکت الکترونیکی که چندین نوع مدارات الکترونیکی مختلف را می‌سازد در نظر گیرید. هر مدار باید از بخش‌های مختلف (از قبیل سوراخ‌کردن، قطعه‌گذاری، آزمایش، و غیره) قبل از بارگیری و ارسال عبور کند. زمان لازم که هر مدار باید از بخش‌های مختلف بگذرد نیز معلوم است. کمترین تعداد مدار در ماه که این شرکت باید تولید کند وجود دارد. البته، شرکت توانائی تولید تعداد بیشتر از کمترین تولید هر مدار را در ماه نیز دارد. سودی که بابت هر مدار شرکت بدست می‌آورد معلوم است. مسئله اینجاست که تعیین کنیم از هر مدار چند عدد تولید شود که در ماه بیشترین سود را شرکت داشته باشد. شبیه این مسئله ممکن است این باشد که کدام یک از این هدف‌ها قیمت تولید خاصی را کمینه می‌کند. این گونه مسائل بهینه‌سازی در این فصل بیشتر بحث خواهد شد.

در اکثر مسائل بهینه‌سازی، تابع هدف f معمولاً بستگی به چندین متغیر، $x_1, x_2, x_3, \dots, x_n$ دارد. این‌ها را متغیرهای کنترول^۴ می‌نامند، زیرا مقادیر آنها را را می‌توان انتخاب کرد. نظریه بهینه‌سازی روش‌هایی برای انتخاب مقادیر بهینه برای متغیرهای کنترول $x_1, x_2, x_3, \dots, x_n$ مدون می‌کند که تابع هدف f را کمینه یا بیشینه کند. در خیلی از حالات، انتخاب مقادیر برای $x_1, x_2, x_3, \dots, x_n$ کاملاً آزاد نبوده بلکه مشروط به محدودیت و قیودی است.

۲.۱۰ مسائل بهینه‌سازی بدون محدودیت

در حساب دیفرانسیل و انتگرال^۵، نشان داده می‌شود که شرط لازم (نه کافی) برای این که f در نقطه P کمینه یا بیشینه داشته باشد این است که مشتق جزئی اول در نقطه P صفر باشد، یعنی،

$$\frac{\partial f}{\partial x_1}(P) = \frac{\partial f}{\partial x_2}(P) = \dots = \frac{\partial f}{\partial x_n}(P) = 0 \quad (1.10)$$

¹object function

²constraints

³control variables

⁴calculus

که در آن نماد (P) نمایشگر مشتق جزئی نسبت به x_i در نقطه P ، یعنی $\left. \frac{\partial f}{\partial x_i} \right|_{x=P}$ است. اگر $n = 1$ و تابع هدف $y = f(x)$ باشد، شرط لازم برای بیشینه یا کمینه (اکسترمم) در نقطه x_0 برابر $y'(x_0) = 0$ است

برای این که y در نقطه x_0 کمینه داشته باشیم، باید $0 < y''(x_0) < 0$ باشد.

برای این که y در نقطه x_0 بیشینه داشته باشیم، باید $0 > y''(x_0) < 0$ باشد.

برای f که شامل چندین متغیر است، شرطی که برای f کمینه نسبی داشته باشد، خیلی مشکل و پیچیده است. اولاً، رابطه (۱.۱۰) باید ارضاء شود. ثانیاً، فرم درجه دوم

$$Q = \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}(P)[(x_i - x_i(P))][(x_j - x_j(P))] \quad (۲.۱۰)$$

باید برای تمام گزینه‌های x_i و x_j در نزدیکی نقطه P مثبت و $Q = 0$ فقط وقتی $x_i = x_i(P)$ برای $i = 1, 2, \dots, n$ باشد. این شرط از بسط سری تیلور^۵ حول نقطه P فقط با استفاده

از جملات $\frac{\partial^2 f}{\partial x_i \partial x_j}(P)$ بدست می‌آید. این میدهد

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f(P) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(P)[x_i - x_i(P)] \\ &\quad + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}(P)[(x_i - x_i(P))][(x_j - x_j(P))] \end{aligned} \quad (۳.۱۰)$$

اگر $i = 1, 2, \dots, n$ برای f دارای کمینه نسبی در نقطه P باشد، در آن صورت $\frac{\partial f}{\partial x_i}(P) = 0$ است. $f(x_1, x_2, \dots, x_n) - f(P) > 0$ است. بنابراین، برای $f(x_1, x_2, \dots, x_n) - f(P) = Q$ که یک کمینه نسبی در نقطه P دارد، Q باید برای تمام گزینه‌های x_j و x_i در نزدیکی P مثبت باشد.

چون این تحلیل وقتی f یک تابع چندین متغیره است کاملاً پیچیده می‌باشد، روش دیگری معمولاً برای حل این مسئله بکار می‌رود. یکی از این روشها، روش تندترین نزول^۶ است. در این روش، ابتداء لازم است برای یک نقطه‌ای که بیشینه یا کمینه وجود دارد حدس زده شود. استفاده از شبکه در محاسبه تابع در مقادیر مختلف متغیرهای کنترول، می‌تواند کمک کند تا نقطه شروع خوبی برای فرآیند تکرار مدون شود.

۳.۱۰ روش تندترین نزول

تابع f با سه متغیر x , y و z را در نظر گیرید. از حساب دیفرانسیل و انتگرال میدانیم که گرادیان f بصورت ∇f نوشته می‌شود و چنین تعریف می‌شود

$$\nabla f = \frac{\partial f}{\partial x} \hat{\mathbf{e}}_x + \frac{\partial f}{\partial y} \hat{\mathbf{e}}_y + \frac{\partial f}{\partial z} \hat{\mathbf{e}}_z$$

در اینجا $\hat{\mathbf{e}}_x$, $\hat{\mathbf{e}}_y$ و $\hat{\mathbf{e}}_z$ بترتیب بردارهای واحد در جهت‌های x , y و z است. در نقطه (x_0, y_0, z_0) ، میدانیم که $\nabla f(x_0, y_0, z_0)$ در جهتی است که سرعت تغییرات f نسبت به

فاصله بیشترین است. بردار واحد $\hat{\mathbf{e}}_g$ که بسوی این جهت می‌باشد برابر است با

$$\hat{\mathbf{e}}_g = \frac{\nabla f}{|\nabla f|}$$

^۵Taylor series expansion

^۶method of steepest descent

در اینجا

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial z}\right)^2}$$

برای پیدا کردن کمینه نسبی از طریق روش تندترین نزول، ما در یک نقطه شروع کرده و با قدمهای کوتاه در جهت تندترین نزول یعنی \hat{e}_g - حرکت می‌کنیم. فرض کنید $(x_{n-1}, y_{n-1}, z_{n-1})$ موقعیت جدید روی تکرار n و (x_n, y_n, z_n) موقعیت قدیم باشد. در این صورت،

$$\begin{aligned} x_{n+1} &= x_n - \frac{\frac{\partial f}{\partial x}(x_n, y_n, z_n)}{|\nabla f(x_n, y_n, z_n)|} \Delta s \\ y_{n+1} &= y_n - \frac{\frac{\partial f}{\partial y}(x_n, y_n, z_n)}{|\nabla f(x_n, y_n, z_n)|} \Delta s \\ z_{n+1} &= z_n - \frac{\frac{\partial f}{\partial z}(x_n, y_n, z_n)}{|\nabla f(x_n, y_n, z_n)|} \Delta s \end{aligned} \quad (4.10)$$

که در آن Δs طول کوچکی است.

مثال ۱.۱۰:

تابع زیر داده شده است.

$$f(x_1, x_2) = 4 + 4.5x_1 - 4x_2 + x_1^2 + 2x_2^2 - x_1x_2 + x_1^4 - 2x_1^2x_2$$

با استفاده از روش تندترین نزول، کمینه f را با شروع از نقطه $(x_1, x_2) = (6, 10)$ بدست آورید. مقدار Δs را در نظر گرفته و حداقل ۱۰۰ تکرار بکار برد.

```
% Example_10_1.m:
% This program determines a relative minimum by the
% method of steepest descent. The function is:
% f(x1,x2) = 4+4.5*x1-4*x2+x1^2+2*x2^2
% -2*x1*x2+x1^4-2*x1^2*x2
% Note: this function has known minima at
% (x1,x2)=(1.941,3.854) and (-1.053,1.028). The
% functional values at the minima points are 0.9855 and
% -0.5134 respectively.
clear; clc;
% Define function and its partial derivatives
fx_func = @(x1,x2) 4+4.5*x1-4*x2+x1^2+2*x2^2 ...
-2*x1*x2+x1^4-2*x1^2*x2;
dfx1_func = @(x1,x2) 4.5+2*x1-2*x2+4*x1^3-4*x1*x2;
dfx2_func = @(x1,x2) -4+4*x2-2*x1-2*x1^2;
% Define stepping parameters
ds=0.1;
max_iterations = 100;
% First guess
x1=6;
x2=10;
fx=fx_func(x1,x2);
% Print headings and initial guess to screen
fprintf('x1 x2 fx \n');
```

```

fprintf('0: %7.4f %7.4f %10.4f \n',x1,x2,fx);
for n=1:max_iterations
% compute partial derivatives
dfx1=dfx1_func(x1,x2);
dfx2=dfx2_func(x1,x2);
% compute magnitude of gradient
gradf_mag=sqrt(dfx1^2+dfx2^2);
% compute next values of x1 and x2 as per Eqn 10.4
x1n=x1-dfx1/gradf_mag*ds;
x2n=x2-dfx2/gradf_mag*ds;
fxn=fx_func(x1n,x2n);
fprintf('%d: %7.4f %7.4f %10.4f \n',...,n,x1n,x2n,fxn);
% if new value is larger than previous,then minimum
% has been found
if(fxn >fx)
    fprintf('A minimum has been found after ');
    fprintf('%d iterations\n\n',n);
    break;
% otherwise,store new values over current ones
% and continue
else
    x1=x1n;
    x2=x2n;
    fx=fxn;
end
end
if i >= max_iterations
    fprintf('Error: no solution found after ');
    fprintf('%d iterations\n',i);
else
    fmin=fx_func(x1,x2);
    fprintf('The relative minimum occurs at ');
    fprintf('x1=% .4f x2=% .4f\n',x1,x2);
    fprintf('The minimum value for f=% .4f \n',fmin);
end
%-----

```

خروجی برنامه

| | x1 | x2 | fx |
|----|--------|---------|----------|
| 0: | 6.0000 | 10.0000 | 683.0000 |
| 1: | 5.9003 | 10.0077 | 622.7192 |
| 2: | 5.8006 | 10.0155 | 566.2528 |
| 3: | 5.7009 | 10.0233 | 513.4609 |
| . | | | |

```

.
.
.
89: 1.9936 4.0234 1.0001
90: 1.9638 3.9279 0.9883
91: 1.9495 3.8289 0.9901
A minimum has been found after 91 iterations
The relative minimum occurs at x1=1.9638 x2=3.9279
The minimum value for f=0.9883.

```

برای پیدا کردن نتایج دقیق‌تر، می‌توانیم برنامه را با مقادیر اولیه تجدیده نظر شده‌ای (از تکرار ۹۱) دوباره اجرا کرده و Δs و Δx کوچکتری بکار ببریم. بر عکس، بجای نقطه شروع دلخواه که در مثال مشخص شده، می‌توانیم ابتداء یک برنامه شبکه‌ای برای تدوین نقطه شروع بکار ببریم. بعلاوه، برنامه شبکه ممکن است مشخص کند که بیش از یک نقطه کمینه نسبی وجود دارد.

مثال ۲.۱۰

```

% Example_10_2.m: This program calculates the values of
% a specified function f(x1,x2) of 2 variables for
% determining a good starting point for the method of
% steepest descent. The range of interest is from
% -10.0 <= x1 <= 10.0 and -10.0 <= x2 <= 10.0.
clear; clc;
% Define function of interest:
fx_func = @(x1,x2) 4+4.5*x1-4*x2+x1^2+2*x2^2 ...
-2*x1*x2+x1^4-2*x1^2*x2;
% Define grid endpoints and step size
x1min=-10.0; x1max=10.0; dx1=2.0;
% Define grid and calculate f(x1,x2) at each point
x1=x1min:dx1:x1max;
x2=x2min:dx2:x2max;
for i=1:length(x1)
for j=1:length(x2)
f(i,j) = fx_func(x1(i),x2(j));
end
end
% Print heading
fprintf('-----');
fprintf('-----');
fprintf('-----\n');
fprintf('x2 | ');
fprintf('x1\n');
fprintf('-----');
fprintf('-----');
fprintf('-----\n');
fprintf(' | ');
for i=1:length(x1)

```

```

fprintf('%7.1f ',x1(i));
end
fprintf('\n');
fprintf('-----');
fprintf('-----');
fprintf('-----\n');
% Print values of f(x1,x2)
for j=1:length(x2)
    fprintf('%6.1f |',x2(j));
    for i=1:length(x1)
        fprintf('%8.1f ',f(i,j));
    end
    fprintf('\n');
end
%-----

```

خروجی برنامه

| x2 | | x1 | | | | | | | | | | |
|-------|--|---------|--------|--------|-------|-------|-------|-------|-------|--------|--------|---------|
| | | -10.0 | -8.0 | -6.0 | -4.0 | -2.0 | 0.0 | 2.0 | 4.0 | 6.0 | 8.0 | 10.0 |
| -10.0 | | 12099.0 | 5488.0 | 2149.0 | 738.0 | 295.0 | 244.0 | 393.0 | 934.0 | 2443.0 | 5880.0 | 12589.0 |
| -8.0 | | 11659.0 | 5184.0 | 1949.0 | 610.0 | 207.0 | 164.0 | 289.0 | 774.0 | 2195.0 | 5512.0 | 12069.0 |
| -6.0 | | 11235.0 | 4896.0 | 1765.0 | 498.0 | 135.0 | 100.0 | 201.0 | 630.0 | 1963.0 | 5160.0 | 11565.0 |
| -4.0 | | 10827.0 | 4624.0 | 1597.0 | 402.0 | 79.0 | 52.0 | 129.0 | 502.0 | 1747.0 | 4824.0 | 11077.0 |
| -2.0 | | 10435.0 | 4368.0 | 1445.0 | 322.0 | 39.0 | 20.0 | 73.0 | 390.0 | 1547.0 | 4504.0 | 10605.0 |
| 0.0 | | 10059.0 | 4128.0 | 1309.0 | 258.0 | 15.0 | 4.0 | 33.0 | 294.0 | 1363.0 | 4200.0 | 10149.0 |
| 2.0 | | 9699.0 | 3904.0 | 1189.0 | 210.0 | 7.0 | 4.0 | 9.0 | 214.0 | 1195.0 | 3912.0 | 9709.0 |
| 4.0 | | 9355.0 | 3696.0 | 1085.0 | 178.0 | 15.0 | 20.0 | 1.0 | 150.0 | 1043.0 | 3640.0 | 9285.0 |
| 6.0 | | 9027.0 | 3504.0 | 997.0 | 162.0 | 39.0 | 52.0 | 9.0 | 102.0 | 907.0 | 3384.0 | 8877.0 |
| 8.0 | | 8715.0 | 3328.0 | 925.0 | 162.0 | 79.0 | 100.0 | 33.0 | 70.0 | 787.0 | 3144.0 | 8485.0 |
| 10.0 | | 8419.0 | 3168.0 | 869.0 | 178.0 | 135.0 | 164.0 | 73.0 | 54.0 | 683.0 | 2920.0 | 8109.0 |

چون مقدار تابع در $(x_1, x_2) = (0, 2)$ کمتر از مقادیر نقاط اطراف است، کمینه نسبی در نزدیکی $(x_1, x_2) = (0, 2)$ مشکوک است. کمینه نسبی دوم در نزدیکی $(x_1, x_2) = (2, 4)$ ظاهر میشود.

۴.۱۰ تابع fminunc در متلب

متلب تابع `fminunc(FUN, X0)` را برای حل مسائل کمینه‌سازی بدون محدودیت تهیه کرده است. دو آرگومان `fminunc` کمینه کردن تابع (`FUN`) و حدس اولیه (`X0`) است. تابع (`FUN`) را می‌توان بصورت یک تابع جداگانه فایل *m*. تعريف کرد و یا ممکن است در متن برنامه بی‌نام آنرا نوشت. خروجی `fminunc` کمینه‌ای است که به حدس اولیه نزدیکتر است. توجه کنید که متلب تابع دیگری که بیشینه را محاسبه کند ندارد. برای پیدا کردن بیشینه، تابع خود را دوباره تعريف کرده که مقدار منفی بدهد و سپس از `fminunc` برای پیدا کردن کمینه استفاده کنید.

مثال ۳.۱۰

تابع زیر داده شده است

$$y(x) = x^3 + 40x^2 - 200x + 12$$

بیشینه و کمینه آنرا بدست آورید

```
% Example_10_3.m: Find the minima and maxima of
% y = x^3 + 40x^2 -200x + 12
clc; clear;
% First, plot the function so that we can make some
% useful initial guesses:
x = -40:40;
for i=1:length(x)
y(i) = x(i)^3 + 40*x(i)^2 -200*x(i) + 12;
end
plot(x,y);
xlabel('x'), ylabel('y'), grid, title('y vs x');
% Next, find the minimum and maximum using MATLAB's
% anonymous function method (find first guesses by
% eyeballing plot).
[xmin,minvalue] = fminunc(@(x) x.^3+40*x.^2-200*x+12,2 );
% Note: To find a maximum, instead find the minimum of
% the negative of the function.
[xmax,maxvalue] = fminunc(@(x) -(x.^3+40*x.^2-200*x+12),...
-30 );
% Print results
fprintf('xmin=% .1f minvalue=% .1f ', xmin,minvalue);
fprintf('xmax=% .1f maxvalue=% .1f\n', xmax,-maxvalue);
%-----
```

خروجی برنامه

xmin = 2.3 minvalue = -224.2, xmax = -29.0 maxvalue = 15063.0

مقادیر کمینه و بیشینه y را تعیین کرده و نتایج را با نتایج تابع $fminunc$ مقایسه نماید.

۵.۱۰ بهینه‌سازی با محدودیت

در خیلی از مسائل بهینه‌سازی، متغیرهای تابعی که باید کمینه یا بیشینه شوند تمام آنها مستقل نیستند بلکه بیک یا چند شرط محدود و مقید هستند. یک مثال ساده که مفهوم محدودیت را روشن کند در زیر آورده می‌شود.

مثال ۴.۱۰:

فرض کنید میخواهیم مقادیر بیشینه یا کمینه تابع هدفی بصورت زیر را

$$(5.10) \quad f(x, y) = 2x + 3y$$

با محدودیت‌های زیر تعیین کنیم:

- باندهای پائینی $(LB) : x \geq 0, y \geq 0$

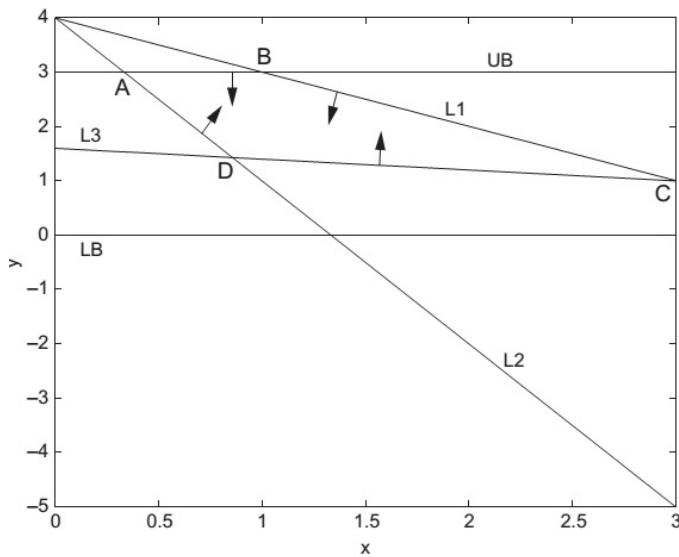
- باندهای بالائی $(UB) : x \leq 3, y \leq 3$

• محدودیت اول $(L1) : x + y \leq 4$

• محدودیت دوم $(L2) : 6x + 2y \geq 8$

• محدودیت سوم $(L3) : x + 5y \geq 8$

باندها در شکل (۱.۱۰) نشان داده شده‌اند. کد متلبی که شکل (۱.۱۰) را تولید می‌کند بدین قرار است. حروف در نقطه تقاطع با استفاده از وارد کردن گزینه **Y** روی شکل متلب بدست می‌آید.



شکل ۱.۱۰: ناحیه مجاز برای کنترول متغیرها در تعیین نقاط کمینه و بیشینهتابع هدف.

```
% Example_10_4.m
% To draw lines, specify two points on the line
% upper bound on y
x(1)=0; y(1)=3; x(2)=3; y(2)=3;
% upper bound on x
x1(1)=3; y1(1)=0; x1(2)=3; y1(2)=3;
% lower bound on y
x2(1)=0; y2(1)=0; x2(2)=3; y2(2)=0;
% lower bound on x
x3(1)=0; y3(1)=0; x3(2)=0; y3(2)=3;
% line L1
x4(1)=0; y4(1)=4; x4(2)=3; y4(2)=1;
% line L2
x5(1)=0; y5(1)=4; x5(2)=3; y5(2)=-5;
% line L3
x6(1)=0; y6(1)=8/5; x6(2)=3; y6(2)=1;
% plot the allowable region
```

Yinsert option

```
plot(x,y,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6)
xlabel('x'), ylabel('y'), title('y vs x');
```

برای هر محدودیت، ما بجای نامساوی یک علامت مساوی رسم خطوط برای L_1 ، L_2 ، و L_3 قرار میدهیم و سپس یک پیکان برای ناحیه مربوط به نامساوی اضافه می‌کنیم. این ناحیه مجازی که مقادیر بیشینه و کمینه را برای تابع f در نظر است، مشخص می‌کند.

در شکل ناحیه مجاز برای (x, y) در تعیین مقادیر بیشینه و کمینه تابع f در ناحیه چند ضلعی $ABCD$ مشاهده می‌کنیم. برای تابع هدف مشخص، مقدار کمینه f روی خط L_3 (بین نقاط D و C) یا روی خط L_2 (بین نقاط A و D) است. می‌توانیم این را مشاهده کنیم اگر نقطه P را روی خط L_3 و نقطه Q مستقیماً بالای آن قرار دهیم؛ در این صورت مقادیر x در نقاط P و Q یکسان خواهد بود، اما مقدار y در نقطه Q بیشتر از y در نقطه P خواهد بود. بنابراین، f در نقطه Q بیشتر از f در نقطه P خواهد بود.

از اینرو، مقدار کمینه f یا روی خط L_3 یا روی خط L_2 است.

معادله خط L_3 برابر است با

$$y = \frac{8}{5} - \frac{x}{5} \quad (\text{خط } L_3)$$

با قرار دادن این رابطه برای y در معادله (۵.۱۰) مقدار f را روی خط L_3 میدهد، که هست

$$y = \frac{24}{5} + \frac{7x}{5} \quad (\text{روی } L_3)$$

مقدار کمینه برای f وقتی اتفاق می‌افتد که مقدار مجاز x روی L_3 کمترین باشد، که آن در نقطه D است. نقطه D از تقاطع خط L_3 و L_2 نتیجه می‌شود. معادله خط L_2 چنین است

$$y = 4 - 3x \quad (\text{خط } L_2)$$

مختصات x در نقطه D را می‌توان با مساوی قرار دادن رابطه y برای خط L_1 و خط L_2 ، مقدار x را در نقطه D ، یا $x_D = \frac{12}{14} = 0.8571$ می‌دهد. در نقطه D ، مقدار f برابر 6.00 است.

با قرار دادن رابطه y روی خط L_2 در معادله (۵.۱۰) معادله‌ای برای f روی خط L_2 میدهد، که هست

$$y = 12 - 7x \quad (\text{روی خط } L_2)$$

مقدار کمینه برای f روی خط AD وقتی اتفاق می‌افتد که بیشینه مجاز مقدار x روی AD باشد، که آن نقطه D است. بنابراین، در ناحیه مجاز $f_{min} = 6.000$ است.

از شکل (۱۱.۰)، مشاهده می‌کنیم که مقدار بیشینه f یا در نقطه B یا جائی دیگر روی خط L_1 خواهد بود. معادله خط L_1 برابر است با

$$y = 4 - x \quad (\text{روی خط } L_1)$$

با قرار دادن این رابطه برای y در معادله (۵.۱۰) خواهیم داشت

$$y = 12 - x \quad (\text{روی خط } L_1)$$

بیشینه f جائی اتفاق می‌افتد که x روی خط L_1 کمینه بوده و آن در نقطه B است. در نقطه B ، $x = 1$ است. بنابراین، $f_{max} = 11.0$ می‌باشد.

۶.۱۰ ضرب‌کننده‌های لاغرانژ

بحث خیلی کلی و ریاضی مسائل بهینه‌سازی با محدودیت بقرار زیر است. فرض کنید تابع هدفی بصورت $f(x_1, x_2, x_3, \dots, x_n)$ با متغیرهای $x_1, x_2, x_3, \dots, x_n$ تحت محدودیت داده شده است،

$$\begin{aligned} \Phi_1(x_1, x_2, x_3, \dots, x_n) &= 0 \\ \Phi_2(x_1, x_2, x_3, \dots, x_n) &= 0 \\ &\vdots \\ \Phi_N(x_1, x_2, x_3, \dots, x_n) &= 0 \end{aligned} \quad (6.10)$$

از نظر تئوری، Nx ها را می‌توان بر حسب x های مانده حل کرد. در این صورت N متغیر را می‌توان از تابع هدف f با جایگزاري حذف کرد، و مسئله نهائی را می‌توان بنحوی که گوئی هیچگونه محدودیتی ندارد حل کرد. این روش را روش ضمنی^۱ می‌نامند و بیشتر حالات‌ها عملی نیست.
روش ضرب کننده‌های لاغرانژ^۲ و سیلهای برای حل مسائل با تحلیل محدودیت است. فرض کنید می‌خواهیم $f(x_1, x_2, x_3, \dots, x_n)$, $Phi_1(x_1, x_2, x_3, \dots, x_n)$, $Phi_2(x_1, x_2, x_3, \dots, x_n)$, ..., $Phi_n(x_1, x_2, x_3, \dots, x_n)$ مانند رابطه (۶.۱۰) بیشینه شود. تابع لاغرانژ F بصورت زیر تعریف می‌شود

$$F(x_1, x_2, x_3, \dots, x_n) = f(x_1, x_2, x_3, \dots, x_n) + \lambda_1 \Phi_1(x_1, x_2, x_3, \dots, x_n) \\ + \lambda_2 \Phi_2(x_1, x_2, x_3, \dots, x_n) + \dots + \lambda_N \Phi_N(x_1, x_2, x_3, \dots, x_n)$$

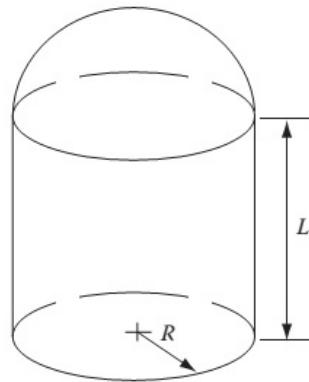
که در آن λ_i در ضرب کننده لاغرانژ مجهول است و باید تعیین شود. اکنون می‌نویسیم

$$\frac{\partial F}{\partial x_1} = 0, \quad \frac{\partial F}{\partial x_2} = 0, \quad \frac{\partial F}{\partial x_3} = 0, \quad \dots, \quad \frac{\partial F}{\partial x_n} = 0 \\ \Phi_1 = 0, \quad \Phi_2 = 0, \quad \Phi_3 = 0, \quad \dots, \quad \Phi_N = 0 \quad (7.10)$$

(تجه: $\frac{\partial F}{\partial \lambda_j} = 0$ بدین معنی است که $\Phi_j = 0$ میباشد).
این دستگاه $n + N$ معادله تمام بیشینه‌ها یا کمینه‌های f را میدهد [۱۱]

مثال ۵.۱۰:

یک انبار غله (سیلو) شامل یک استوانه مدور بشعاع R و ارتفاع L با یک نیم‌گره سقف آن است (به شکل ۲.۱۰ مراجعه کنید). فرض کنید که سیلو باید دارای حجم $V = 8400 m^3$ باشد. ابعاد R و L را بنحوی تعیین کنید که سطح سیلو S کمینه باشد. فرض کنید سطح قاعده سیلو از جنس بدن است. توجه کنید که $S_{sphere} = 4\pi R^2$ و $V_{sphere} = \frac{4}{3}\pi R^3$ است. حل:



شکل ۲.۱۰: یک انبار غله شامل یک استوانه مدور با سقفی نیم کره.

$$V = \frac{2}{3}\pi R^3 + \pi R^2 L \quad (8.10)$$

$$\Phi = \frac{2}{3}\pi R^3 + \pi R^2 L - V = 0 \quad (9.10)$$

$$S = 2\pi RL + \pi R^2 + 2\pi R^2 = 2\pi RL + 3\pi R^2 \quad (10.10)$$

^۱implicit method

^۲Lagrange multipliers

در این حالت، سطح S باید کمینه باشد و حجم V محدود است. تابع لاگرانژ برابر است با

$$\begin{aligned} F &= S + \lambda\Phi \\ &= 2\pi RL + 3\pi R^2 + \lambda \left(\frac{2}{3}\pi R^3 + \pi R^2 L - V \right) \end{aligned} \quad (11.10)$$

متغیرها R ، L و λ هستند. با گرفتن مشتق نسبت به R و L خواهیم داشت

$$\frac{\partial F}{\partial R} = 2\pi L + 6\pi R + \lambda(2\pi R + 2\pi RL) = 0 \quad (12.10)$$

$$\frac{\partial F}{\partial L} = 2\pi R + \lambda\pi R^2 = 0 \quad \rightarrow \quad \lambda R = -2 \quad \text{یا} \quad \lambda = -\frac{2}{R} \quad (13.10)$$

با قرار دادن λ از رابطه (13.10) در رابطه (12.10) داریم

$$2\pi L + 6\pi R - \frac{2}{R}(2\pi R^2 + 2\pi RL) = 0 \quad (14.10)$$

رابطه (14.10) بصورت $R = L - R = 0$ یا $R = L$ خلاصه می‌شود.

با قرار دادن این نتیجه در رابطه (A.10) خواهیم داشت

$$V = \frac{2}{3}\pi R^3 + \pi R^3 = \frac{5}{3}\pi R^3$$

برای $V = 8400 m^3$

$$R = \left(\frac{8400 \times 3}{5\pi} \right)^{1/3} = 11.7065 m$$

با گذاردن مقادیر R و L در رابطه (10.10) خواهیم داشت

۷.۱۰ تابع fmincon متلب

تابع متلب برای حل مسائل بهینه‌سازی با محدودیت fmincon است. تابع آرگومانهای تحت نام کاربر FUN، نقطه شروع X0، بعلاوه آرگومانهای که بستگی به نوع محدودیت در مسئله تعیین شده، قبول می‌کند.

تابع FUN تابع هدفی که باید بهینه شود را تعیین می‌کند. بردار X0 حدس اولیه کنترول متغیرهای که تابع هدف را کمینه می‌نماید است. تابع fmincon متلب چهار نوع محدودیت را بکار می‌برد:

۱. محدودیت‌های نامساوی: محدودیتها شامل یک یا چند نامساوی بصورت $AX \leq B$ است. فرض کنید مسئله سه محدودیت نامساوی خطی بصورت زیر مشخص می‌کند

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ a_{31}x_1 + a_{32}x_2 + \cdots + a_{3n}x_n &\leq b_3 \end{aligned} \quad (15.10)$$

این محدودیتها در متلب برای استفاده در fmincon بصورت زیر مشخص می‌شوند:

$$A = [a_{11} \ a_{12} \ \cdots a_{1n}; \ a_{21} \ a_{22} \ \cdots a_{2n}; \ a_{31} \ a_{32} \ \cdots a_{3n}]$$

$$B = [b_1; \ b_2; \ b_3]$$

۲. محدودیت‌های تساوی: محدودیتها شامل یک یا چند تساوی بصورت $A_{eq}X = B_{eq}$ است. فرض

کنید مسئله دو محدودیت تساوی بصورت زیر مشخص می‌کند.

$$\tilde{a}_{11}x_1 + \tilde{a}_{12}x_2 + \cdots + \tilde{a}_{1n}x_n = \tilde{b}_1$$

$$\tilde{a}_{21}x_1 + \tilde{a}_{22}x_2 + \cdots + \tilde{a}_{2n}x_n = \tilde{b}_2$$

که در آن \tilde{a}_{ij} و \tilde{b}_i بترتیب عناصر A_{eq} و B_{eq} هستند. این محدودیتها در متلب برای استفاده در fmincon بصورت زیر مشخص می‌شوند:

$$A_{eq} = [\tilde{a}_{11} \ \tilde{a}_{12} \ \dots \ \tilde{a}_{1n}; \tilde{a}_{21} \ \tilde{a}_{22} \ \dots \ \tilde{a}_{2n}]$$

$$B_{eq} = [\tilde{b}_1; \tilde{b}_2]$$

۳. کرانه‌ها: محدودیت شامل حدود بالا یا پائین است که بصورت $X \geq LB$ و $UB \leq X$ عنوان می‌شوند. در این صورت، شما در متلب کرانه‌هایی برای استفاده در fmincon بصورت زیر مشخص می‌کنید.

$$LB = [l_1 \ l_2 \ \dots \ l_n]$$

$$UB = [u_1 \ u_2 \ \dots \ u_n]$$

که در آن $x_1 \leq u_1, x_2 \leq u_2, \dots, x_n \leq u_n$ و $x_1 \geq l_1, x_2 \geq l_2, \dots, x_n \geq l_n$ است.

۴. محدودیت‌های غیرخطی: این محدودیتها باتابع $[C, C_{eq}] = \text{NONLCON}(X)$ تعریف می‌شوند. در اینجا C یک تابع متلب بوده که محدودیت نامساوی غیرخطی C و محدودیت تساوی C_{eq} را مشخص می‌کند. فرض کنید مسئله دو محدودیت نامساوی غیرخطی و یک محدودیت تساوی غیرخطی مشخص می‌کند. محدودیت نامساوی غیرخطی و محدودیت تساوی غیرخطی توابع f_1, f_2 و f_3 لازم است بنحوی برپا و مشخص شوند

$$f_1(x_1, x_2, \dots, x_n) \leq 0, \quad f_2(x_1, x_2, \dots, x_n) \leq 0, \quad f_3(x_1, x_2, \dots, x_n) \leq 0$$

در این صورت در تابع NONLCON روابط $C(2) = f_2(x_1, x_2, \dots, x_n)$ ، $C(1) = f_1(x_1, x_2, \dots, x_n)$ و $C_{eq}(1) = f_3(x_1, x_2, \dots, x_n)$ قرار میدهیم.

توضیح خیلی از احصارهای fmincon را می‌توان با نوشتن help fmincon در پنجره فرمان متلب بدست آورد. برخی از آنها بعداً توضیح داده خواهد شد.

کاربرد اول

```
X = FMINCON(FUN,X0,A,B)
starts at X0 and finds a minimum X
to the function FUN, subject to the linear inequalities A*X <
= B. FUN accepts input X and returns a scalar function value F
evaluated at X. X0 may be a scalar, vector, or matrix.
```

اولین نسخه فقط وقتی که تمام محدودیتها نامساوی خطی هستند بکار می‌رود؛ یعنی، $A*X \leq B$ در این حالت، متن در $X0$ شروع شده و متغیرهای کنترول X را تابع هدف که در FUN است را کمینه کرده و مقادیر X را به برنامه‌ای که آنرا می‌خواند بر می‌گرداند. حدس اولیه $X0$ یک بردار ستونی است، و A ، B یک دستگاه معادله که محدودیت نامساوی را تعریف می‌کند، توصیف می‌کند.

کاربرد دوم

```
X = FMINCON(FUN,X0,A,B,Aeq,Beq)
minimizes FUN subject to
the linear equalities Aeq*X = Beq as well as A*X <= B. (Set
A = [] and B = [] if no inequalities exist.)
```

نسخه دوم وقتی بکار می‌رود که محدودیتهای مساوی خطی وجود داشته باشد؛ یعنی، $Aeq*X = Beq$ که در آن Aeq و Beq یک دستگاه از معادلاتی که محدودیتهای مساوی را معلوم میدارند توصیف می‌کند. در این صورت استفاده از نماد $[]$ برای A و B است.

کاربرد سوم

`X = FMINCON(FUN,X0,A,B,Aeq,Beq,LB,UB)` defines a set of lower and upper bounds on the design variables, X , so that a solution is found in the range $LB \leq X \leq UB$. Use empty matrices for LB and UB if no bounds exist. Set $LB(i) = -\infty$ if $X(i)$ is unbounded below; set $UB(i) = \infty$ if $X(i)$ is unbounded above.

نسخه سوم وقتی بکار می‌رود که می‌خواهیم حدود بالا و پائین LB و UB در متغیرهای کنترول را تنظیم کند. در اینجا LB و UB بردارهایی هستند که شامل قیود روی X می‌باشند.

کاربرد چهارم

`X = FMINCON(FUN,X0,A,B,Aeq,Beq,LB,UB,NONLCON)` subjects the minimization to the constraints defined in function `NONLCON`. The function `NONLCON` accepts X and returns the vectors C and Ceq , representing the nonlinear inequalities and equalities respectively. `FMINCON` minimizes `FUN` such that $C(X) \leq 0$ and $Ceq(X) = 0$. (Set $LB = []$ and/or $UB = []$ if no bounds exist, do the same for A, B, Aeq, Beq if there are no linear inequality or linear equality constraints.)

نسخه چهارم وقتی استفاده می‌شود که محدودیت‌های غیرخطی به صورت تابع `NONLCON` از X مانند آنچه در مثال‌های ۷.۱۰ و ۸.۱۰ نمایش داده شده، تعریف می‌کند.

توجه کنید که در تمام این کاربردها، ما `fmincon` را بعنوان یک بردار تکی یعنی، از (\dots) فراخواندیم. البته، برای راحتی، می‌توانیم نیز از (\dots , $X, FVAL$) = `fmincon(FUN, X0, ..., FVAL)` استفاده کنیم. که آنهم `FVAL` را که مقدار کمینه تابع هدف است، بر می‌گرداند.

مثال ۶.۱۰

در این مثال، ما از تابع `fmincon` برای تعیین مقادیر بیشینه و کمینه تابع هدف مثال ۴.۱۰ تحت محدودیت ویژه این مثال، استفاده می‌کنیم.

```
% Example_10_6.m: Linear programming example using
% MATLAB's fmincon function for optimization
% Object function: 2*x(1)+3*x(2)
% Lower bounds
% x(1)>=0 x(2)>=0
% Upper bounds
% x(1)<=3 x(2)<=3
% Inequality constraints
% x(1)+x(2)<=4
% 6*x(1)+2*x(2)>=8
% x(1)+5*x(2)>=8
clear; clc;
lb=[0; 0]; % Lower bound
ub=[3; 3]; % Upper bound
x0=[0; 0]; % Initial guess
```

```

Aeq=[]; % No linear equality constraints
Beq=[]; % No linear equality constraints
A=[1 1; -6 -2; -1 -5]; % Linear inequality constraints
B=[4; -8; -8]; % Linear inequality constraints
[xmin,fvalmin] = fmincon(@object_fmin, ...
x0,A,B,Aeq,Beq,lb,ub);
[xmax,fvalmax] = fmincon(@object_fmax, ...
x0,A,B,Aeq,Beq,lb,ub);
% Print min and max values:
fprintf('MIN\n');
fprintf('-----\n');
fprintf('xmin(1)=%.4f xmin(2)=%.4f \n',xmin(1),xmin(2));
fprintf('fmin=%.4f \n\n',fvalmin);
fprintf('MAX\n');
fprintf('-----\n');
fprintf('xmax(1)=%.4f xmax(2)=%.4f \n',xmax(1),xmax(2));
fprintf('fmax=%.4f \n',-fvalmax);
% object_fmin.m
% This function works with Example_10_6.m
function f = object_fmax(x)
f = -(2*x(1)+3*x(2));
%-----

```

خروجی برنامه

```

MIN
-----
xmin(1) = 0.8571 xmin(2) = 1.4286
fmin = 6.0000
MAX
-----
xmax(1) = 1.0000 xmax(2) = 3.0000
fmax = 11.0000

```

مثال ۷.۱۰

با استفاده از تابع `fmincon` سطح کمینه انبار کنید (سیلو) مثال ۷.۱۰ را حل کنید.

```

% Example_10_7.m: This program minimizes the material
% surface area of a silo. The silo consists of a right
% cylinder topped by a hemisphere. Solve where the silo
% volume is 8400 m^3.
% Define the vector X = [R,L] where R is radius and L is
% length (as drawn in Figure 10.1)
% The function objfun_silo calculates the silo surface
% area and is the objective function to be minimized.
% The function confun_silo defines the constraint that

```

```
% the silo volume be fixed at 8400.
clear; clc;
global V;
V=8400;
% Define bounds: radius and length must be positive.
LB = [0,0];
UB = [];
% Initial guesses for R and L:
Xo = [10.0 20.0];
fprintf('This program minimizes the surface area\n');
fprintf('of a silo consisting of a right cylinder\n');
fprintf('topped by a hemisphere. The silo volume\n');
fprintf('is fixed at %.0f m^3.\n',V);
% Run the optimization. We have no linear constraints,
% so pass [] for those arguments:
[X,fval]=fmincon(@objfun_silo,Xo, ...
[],[],[],LB,UB,@confun_silo);
% Print results:
fprintf('Silo volume: %9.3f m^3\n',V);
fprintf('Minimum surface area: %9.3f m^2\n',fval);
fprintf('Optimum radius: %9.3f m\n',X(1));
fprintf('Optimum length: %9.3f m\n',X(2));
-----
% objfun_silo.m (object function for Example 10.7)
function SA = objfun_silo(X)
% Calculate surface area of silo. The variables are:
% radius R = X(1),length of cylinder L = X(2)
SA = 2.0*pi*X(1)*X(2) + 3.0*pi*X(1)^2;
-----
% confun_silo.m (constraint function for Example 10.7)
function [c,ceq] = confun_silo(X)
% Variables are:
% radius R = X(1),length of cylinder L = X(2).
global V;
% Nonlinear equality constraints:
ceq = pi*X(1)^2*X(2) + 2.0/3.0*pi*X(1)^3 -V;
% No nonlinear inequality constraints:
c = [];
-----
```

خروجی برنامه

Optimization Problem:

This program minimizes the surface area of a silo
consisting of a right cylinder topped by a hemisphere.
The silo volume is set at 8400 m³

```
Silo volume: 8400.000 m^3
Minimum surface area: 2152.651 m^2
Optimum radius: 11.707 m
Optimum length: 11.706 m
```

مثال ۸.۱۰:

فرض کنید مثال ۷.۱۰ را معکوس کرده و بیشینه حجم سیلو مثال ۷.۱۰ مورد نظر است. اما محدودیت سطح سیلو است که باید کمتر از مقدار مشخص m^2 2152.6 باشد. جواب مسئله با استفاده ازتابع fmincon متلب بقرار زیر است:

```
% Example_10_8.m: This program maximizes the volume of a
% silo. The silo consists of a right cylinder topped by a
% hemisphere. The variables to optimize are radius 'R'
% and cylinder height 'L'.
% Let X(1) = R and X(2) = L.
% Eqn for surface area: S=3*pi*X(1)^2+2*pi*X(1)*X(2);
% Constraint: 3*pi*X(1)^2+2*pi*X(1)*X(2) <= 2152.6 m^2.
% We need to set up the constraint in the form:
% 3*pi*X(1)^2+2*pi*X(1)*X(2)-2152.6 <= 0
% We wish to maximize the volume under the constraint.
% The equation for the volume,V,of the silo is given by
% V=2.0/3.0*pi*X(1)^3 + pi*X(1)^2*X(2).
% Since we wish to maximize V,we will need to place a
% minus sign in front of the expression for V and then
% minimize.
clear; clc;
global SAmatrix;
SAmatrix = 2152.6;
% Define bounds: radius and length must be positive.
LB = [0,0];
UB = [];
% Take an initial guess at the solution
Xo = [10.0 20.0];
fprintf('This program maximizes the volume of a silo\n');
fprintf('consisting of a right cylinder topped by a\n');
fprintf('hemisphere. The maximum surface area is\n');
fprintf('fixed at %.1f m^3.\n',SAmatrix);
% Run the optimization. We have no linear constraints,
% so pass [] for those arguments:
[X,fval]=fmincon(@objfun_silo2,...
Xo,[],[],[],LB,UB,@confun_silo2);
SA = 3*pi*X(1)^2 + 2*pi*X(1)*X(2);
% Print results:
fprintf('Maximum surface area: %9.3f m^2\n',SA);
fprintf('Maximum volume: %9.3f m^3\n',-fval);
```

```

fprintf('Optimum radius: %9.3f m\n',X(1));
fprintf('Optimum length: %9.3f m\n',X(2));
%-----
% objfun_silo2.m (object function for Example 10.8)
function V = objfun_silo2(X)
% Compute volume of silo. The variables are:
% radius R = X(1),length of cylinder L = X(2)
V = -(2.0/3.0*pi*X(1)^3 + pi*X(1)^2*X(2));
%-----
% confun_silo2.m (constraint function for Example 10.8)
function [c,ceq] = confun_silo2(X)
global SAmmax;
% Variables are:
% radius R = X(1),length of cylinder L = X(2)
% Nonlinear inequality constraint:
c = 3*pi*X(1)^2 + 2*pi*X(1)*X(2) -SAmmax;
% No nonlinear equality constraints:
ceq = [];
%-----

```

خروجی برنامه

Optimization Problem:
This program maximizes the volume of a silo
consisting of a right cylinder topped by a hemisphere.
The maximum surface area is set at 2152.6 m³
Maximum surface area: 2152.600 m²
Maximum volume: 8399.701 m³
Optimum radius: 11.706 m
Optimum length: 11.706 m

مثال ۹.۱۰

دو کارگاه ماشین سازی، ماشین سازی A و ماشین سازی B دو نوع شفت موتور، شفت S1 و شفت S2 تولید می‌کنند. هر کارگاه ماشین سازی دارای دو ماشین تراش است: ماشین تراش T1 و ماشین تراش T2. جدول زیر زمان تولید را برای هر شفت روی هر ماشین و هر مکانی نشان می‌دهد.

| زمان ساخت (دقیقه) | | | | | |
|-------------------|-----|---------|-----|----------------|----------------|
| B ماشین | | A ماشین | | | |
| S2 | شфт | S1 | شфт | S2 | شфт |
| 8 | 5 | 9 | 4 | T ₁ | |
| 5 | 3 | 6 | 2 | | T ₂ |

شфт S1 را 35 دلار و شفت S2 را 85 دلار میفروشند. تعداد شفت‌های S1 و S2 را تعیین کنید که هر کارگاه ماشین سازی و هر ماشین تراش تولید کند که برای یک ساعت بیشترین منافع کسب شود.

```
% Example_10_9.m: Shaft Production Problem
% This program maximizes the revenue/hr for the
% production of two types of shafts, type S1 and type S2.
% There are two machine shops producing these shafts,
% shop A and shop B. Each shop has two types of turning
% machines, T1 and T2, capable of producing these shafts.
% Shop A:
% Machine T1 takes 4 minutes to produce type S1 shafts
% and 9 minutes to produce type S2 shafts.
% Machine T2 takes 2 minutes to produce type S1 shafts
% and 6 minutes to produce type S2 shafts.
% Shop B:
% Machine T1 takes 5 minutes to produce type S1 shafts
% and 8 minutes to produce type S2 shafts.
% Machine T2 takes 3 minutes to produce type S1 shaft
% and 5 minutes to produce type S2 shafts.
% Shaft S1 sells for $35 and shaft S2 sells for $85.
% We wish to determine the number of S1 & S2 shafts that
% should be produced at each shop and by each machine
% that will maximize the revenue/hr for shaft production.
% Let:
% X(1)=# of S1 shafts produced/hr by machine T1 at shop A
% X(2)=# of S2 shafts produced/hr by machine T1 at shop A
% X(3)=# of S1 shafts produced/hr by machine T2 at shop A
% X(4)=# of S2 shafts produced/hr by machine T2 at shop A
% X(5)=# of S1 shafts produced/hr by machine T1 at shop B
% X(6)=# of S2 shafts produced/hr by machine T1 at shop B
% X(7)=# of S1 shafts produced/hr by machine T2 at shop B
% X(8)=# of S2 shafts produced/hr by machine T2 at shop B
% Let r=total revenue/hr for producing these shafts:
% r=35*(X(1)+X(3)+X(5)+X(7))+85*(X(2)+X(4)+X(6)+X(8))
clear; clc;
% Objective function: total revenue per hour for
% manufactured shafts. Have the function return a
% negative number because we are maximizing instead of
% minimizing.
revenue=@(x) -(35*(x(1)+x(3)+x(5)+x(7))+85*(x(2)+x(4)...
+x(6)+ x(8)));
% Take a guess at the solution
Xo = [0 0 0 0 0 0 0 0];
% Lower and upper bounds:
LB = [0 0 0 0 0 0 0 0];
UB = [];
% We have linear inequality constraints. We require that
% each machine make an integral number of shafts per 60
```

```
% minutes,so:
% 4*X(1)+9*X(2) <= 60
% 2*X(3)+6*X(4) <= 60
% 5*X(5)+8*X(6) <= 60
% 3*X(7)+5*X(8) <= 60
A=[4 9 0 0 0 0 0 0;
0 0 2 6 0 0 0 0;
0 0 0 0 5 8 0 0;
0 0 0 0 0 0 3 5];
B=[60 60 60 60]';
% We have no linear equality constraints,so pass [] for
% those arguments.
[X,rmax] = fmincon(revenue,Xo,A,B,[],[],LB,UB);
fprintf('Optimization Results:\n\n');
fprintf('No. of S1 shafts produced at shop A ');
fprintf('by machine T1: %2.0f\n',X(1));
fprintf('No. of S2 shafts produced at shop A ');
fprintf('by machine T1: %2.0f\n',X(2));
fprintf('No. of S1 shafts produced at shop A ');
fprintf('by machine T2: %2.0f\n',X(3));
fprintf('No. of S2 shafts produced at shop A ');
fprintf('by machine T2: %2.0f\n',X(4));
fprintf('No. of S1 shafts produced at shop B ');
fprintf('by machine T1: %2.0f\n',X(5));
fprintf('No. of S2 shafts produced at shop B ');
fprintf('by machine T1: %2.0f\n',X(6));
fprintf('No. of S1 shafts produced at shop B ');
fprintf('by machine T2: %2.0f\n',X(7));
fprintf('No. of S2 shafts produced at shop B ');
fprintf('by machine T2: %2.0f\n\n',X(8));
fprintf('The max revenue per hour: $%.0f /hour\n',-rmax);
%-----
```

خروجی برنامه

Optimization Results:

No. of S1 shafts produced at shop A,T1: 0
 No. of S2 shafts produced at shop A,T1: 7
 No. of S1 shafts produced at shop A,T2: 30
 No. of S2 shafts produced at shop A,T2: 0
 No. of S1 shafts produced at shop B,T1: 0
 No. of S2 shafts produced at shop B,T1: 8
 No. of S1 shafts produced at shop B,T2: 0
 No. of S2 shafts produced at shop B,T2: 12
 The max revenue per hour: \$3274/hour
 %-----

توجه: وقتی برنامه اجرا میشود، متلب یک اختار در صفحه نمایشگر میدهد که میتوان آنرا اغمض و نادیده گرفت. اگر جواب قانون کننده‌ای بدست آمد، متلب بشما میگوید: کمینه محلی پیدا شده و محدودیت‌ها را ارضاء مینماید.

Local minimum found that satisfies the constraints.

مسائل

۱. با استفاده از تابع `fminunc()` متلب، کمینه و بیشینه‌های توابع زیر را بدست آورید.

• الف:

$$f(x) = x^4 + 10x^3 - 20x - 15$$

• ب:

$$f(x) = \frac{\sin^2(x - 0.75)}{x^2 - 1.5x + 0.5625}$$

• ج:

$$f(x) = e^{-0.5x} \sin 2x \quad (x \geq 0) \quad \text{برای}$$

.۲

با استفاده از روش تندترین نزول، موقعیت تقریبی کمینه نسبی f را بدست آورید.

$$f(x_1, x_2) = 8x_1^2 - 20x_1x_2 + 17x_2^2 - 32x_1 + 40x_2$$

از $(x_1, x_2) = (6, 4)$ بعنوان نقطه شروع و $ds = 0.05$ و ۲۰۰ قدم استفاده کنید.

۳. از ضرب کننده لاغرانژ برای تعیین بیشترین حجم جعبه‌ای که میتوان در داخل بیضی‌گون زیر قرار داد، استفاده کرده بطوری که اضلاع آن موازی محورهای مختصات باشند.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

پروژه‌ها

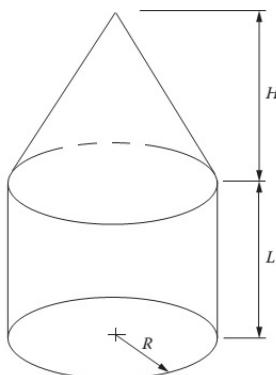
۱.۱۰ پروژه

انبار غله‌ای بشکل یک استوانه مدور است که یک مخروط در بالای آن مطابق شکل (۳.۱۰) قرار دارد. شعاع استوانه و قاعده مخروط R است.

ارتفاع استوانه L و ارتفاع مخروط H است. استوانه، قاعده استوانه و مخروط از یک جنس ساخته شده است. برنامه‌ای با استفاده از تابع `fminunc` متلب نوشته و مقادیر R , H , L را بنحوی تعیین کنید که سطح سیلو برای حجم آن $7000 m^3$ شود.

توجه کنید که حجم و سطح جانبی مخروط از روابط زیر بدست می‌آیند.

$$V = \frac{\pi R^2 H}{3}, \quad S = \pi R \sqrt{R^2 + H^2}$$



شکل ۳.۱۰: سیلو شامل یک استوانه مدور و یک مخروط در بالای آن.

جدول ۱.۱۰: قیمت فروش و هزینه برای هشت نوع کامپیوتر

| نوع کامپیوتر | قیمت فروش (دولار) | هزینه (دولار) |
|--------------|-------------------|---------------|
| C_1 | 675 | 637 |
| C_2 | 805 | 780 |
| C_3 | 900 | 874 |
| C_4 | 1025 | 990 |
| C_5 | 1300 | 1250 |
| C_6 | 1500 | 1435 |
| C_7 | 350 | 340 |
| C_8 | 1000 | 1030 |

۲.۱۰ پروژه

در یک مغازه کامپیوتر فروشی تعدادی کامپیوتر را برای عموم عرصه می‌کند. در آنجا هشت نوع کامپیوتر مختلف در انبارش موجود دارد. جدول (۱.۱۰) اثبات کامپیوتر، قیمت فروش و هزینه هر یک را در مغازه نشان میدهد. فروشنده می‌خواهد هر ماه ۲۰,۰۰۰ دلار از خرید کامپیوتر را ذخیره نماید.

فروشنده می‌خواهد کمتر از 30 درصد هزینه اش را روی کامپیوتر C_1 و C_2 و کمتر از 30 درصد را روی کامپیوتر C_3 و C_4 و کمتر از 10 درصد روی کامپیوتر C_5 و C_6 و کمتر از 30 درصد روی کامپیوتر C_7 و C_8 بگذارد. معازه‌دار حدس می‌زند که می‌تواند 30 درصد نوع C_1 را بیشتر از 20 درصد نوع C_3 را بیشتر از C_4 ، 20 درصد نوع C_5 را بیشتر از C_6 ، و 60 درصد نوع C_7 و C_8 بفروشد. با استفاده ازتابع $fmincon$ متلب، تعداد هر کامپیوتر که بیشترین سود را تدارک می‌کند بدست آورید. تعداد هر نوع کامپیوتر را که می‌بایست در ماه خریداری شود، کل سود در ماه، و کل خرج در ماه فروشگاه را بدست آورید.

۳.۱۰ پروژه

شرکت الکترونیک جونز برای تولید چهار نوع بُرد مدارهای کامپیوتر قراردادی منعقد نموده است. بردّها قبل از ارسال به مقصد، برای مراحل ساخت باید از چهار بخش عبور کنند: مدارات چاپی شامل لایه نشانی و حکاکی (شامل سیم کشی روی برد)، سوراخ کاری؛ قطعه گذاری (نصب کردن ترانزیستورها و میکروپرسسورها...)، و آزمایش کردن کل برد. لازم نیست در حسب دقیقه برای

تولید هر واحد و مقدار سود آن در جدول (۲.۱۰) خلاصه شده است.
هر بخش از تولید سه روز در هفته برای این قرارداد کار می‌کند. کمترین تولید در هفته برای انجام قرار داد در جدول (۳.۱۰) آورده شده است.
برنامه‌ای به زبان متلب نوشته که بتواند:

جدول ۲.۱۰: زمان ساخت برای بردهای مختلف در هر بخش

| بُرد مدار | تهیه مدار چاپی (دقیقه) | سوراخ کردن (دقیقه) | جمع‌بندی (دقیقه) | آزمایش (دقیقه) | سود یک واحد (دولار) |
|-----------|------------------------|--------------------|------------------|----------------|---------------------|
| A بُرد | 15 | 10 | 8 | 15 | 12 |
| B بُرد | 12 | 8 | 10 | 12 | 10 |
| C بُرد | 18 | 12 | 12 | 17 | 15 |
| D بُرد | 13 | 9 | 4 | 13 | 10 |

جدول ۳.۱۰: کمترین تولید هفتگی لازم برای مدارهای مختلف

| بُرد مدار | کمترین تعداد تولید |
|-----------|--------------------|
| A بُرد | 10 |
| B بُرد | 10 |
| C بُرد | 10 |
| D بُرد | 10 |

۱. تعداد هر نوع برد مدار را برای هفته آینده تعیین کنید که بیشترین سود را داشته باشد.
فرض کنید که هشت ساعت در روز ، پنج روز در هفته کارخانه کار می‌کند. توجه: تمام بخش‌ها در یک روز کار نمی‌کنند.
۲. کل سود هفتگی را بدست آورید.
۳. کل زمانی که طول می‌کشد تا تمام بردگاه تولید شود را بر حسب دقیقه تعیین کنید.
۴. کل زمانی که طول می‌کشد تا هر یک از چهار بخش صرف تولید کند را بر حسب دقیقه تعیین کنید.
۵. اطلاعات خواسته شده را در فایلی ثبت کنید.

۴.۱۰ پروژه

کمپانی نفتی XYZ روی سه چاه نفتی (OW1، OW2، OW3) کار کرده و نفت خام برای چهار پالایشگاه (پالایشگاه A، پالایشگاه B، پالایشگاه C، و پالایشگاه D) تولید می‌کند. هزینه حمل نفت خام از هر چاه نفت به هر پالایشگاه، به ظرفیت هر یک از سه چاه نفت، و تقاضای (محدویت کیفیت) برای بنزین هر پالایشگاه در جدول (۴.۱۰) آورده شده است. نفت خام در هر پالایشگاه به شش محصول اصلی تقطیر می‌شود: بنزین، روغن روان‌ساز (روغن ماشین)، نفت منزل، سوخت جت، مازوت (نفت حرارتی)، و پلاستیک. هزینه تقطیر در هر لیتر در هر پالایشگاه از هر چاه نفتی در جدول (۵.۱۰) داده شده است. درصد تولید تقطیر در لیتر در جدول (۶.۱۰) آورده شده است. سود از هر تولید در جدول (۷.۱۰) ثبت شده است.

جدول ۴.۱۰: هزینه حمل در هر صد لیتر به دolar

| ظرفیت چاه | D | C | B | A | پالایشگاه | چاه نفت (لیتر) |
|-----------|------|------|------|------|------------------------|-------------------|
| 7000 | 11 | 10 | 7 | 9 | OW1 | |
| 6100 | 10 | 8 | 10 | 7 | OW2 | |
| 6500 | 7 | 6 | 11 | 10 | OW3 | |
| | 1900 | 2100 | 1800 | 2000 | تقاضای بنزین (لیتر) | |

جدول ۵.۱۰: هزینه تقطیر در هر صد لیتر به دolar

| D | C | B | A | چاه نفت | پالایشگاه |
|----|----|----|----|---------|-----------|
| 14 | 12 | 16 | 15 | OW1 | |
| 10 | 14 | 12 | 17 | OW2 | |
| 17 | 16 | 15 | 12 | OW3 | |

با استفاده از تابع `fmincon` متلب، مقدار لیتری که هر چاه باید تولید کند و به هر یک از چهار پالایشگاه ارسال شود تا بنزین مورد تقاضا را ارضاء کرده و بیشترین سود عاید شود را بدست آورید. اقلام زیر را چاپ کنید.

۱. مقدار نفت تولید شده در هر چاه بر حسب لیتر.
۲. بنزین در هر پالایشگاه بر حسب لیتر.
۳. کل هزینه ترابری و تقطیر تمام تولیدات.
۴. کل عایدی حاصل از فروش تمام فرآوردهای نفتی.
۵. کل سود از تمام تولیدات.

۵.۱۰ پروژه

یک سیستم کنترول درجه دوم دارای پاسخ فرکانسی زیر است.

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (16.10)$$

که در آن ω_n فرکانس طبیعی (بر حسب رادیان بر ثانیه)، و ζ ضریب میرائی^{۱۰} است. پاسخ پله‌واحد^{۱۱} سیستم بعنوان پاسخ حوزه زمانی به پله‌واحد ورودی تعریف می‌شود. در حوزه s ، این به حاصل ضرب $H(s)U(s)$ مربوط می‌شود. در اینجا $U(s) = \frac{1}{s}$ تبدیل لاپلاس تابع پله‌واحد $u(t)$ است. برای پیدا کردن پاسخ پله‌واحد حوزه زمان ($y(t)$ ، عکس تبدیل لاپلاس $H(s)U(s)$) بدست می‌آوریم:

$$y(t) = L^{-1}[H(s)U(s)] = L^{-1}\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \times \frac{1}{s} \quad (17.10)$$

با انجام بسط کسرهای جزئی روی رابطه (۱۷.۱۰) و عکس تبدیل لاپلاس گرفتن از آن توسط

^{۱۰} damping factor

^{۱۱} step response

جدول ۶.۱۰: تولیدهای تقطیر در هر لیتر نفت خام

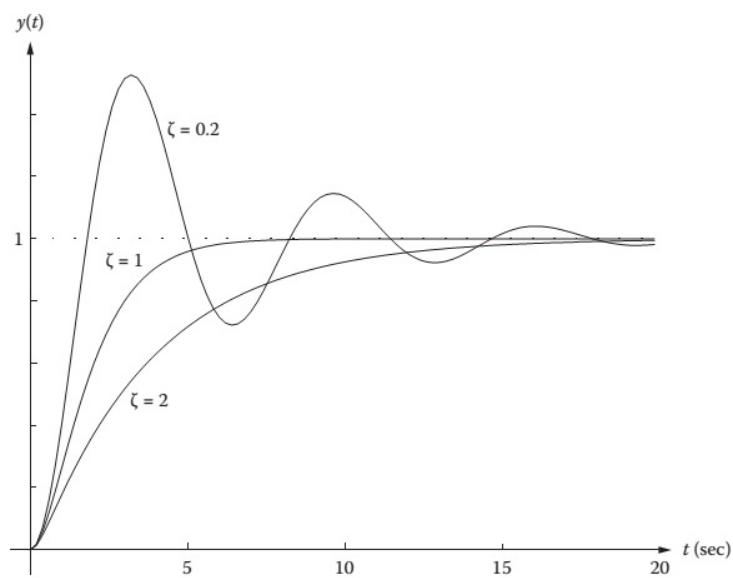
| درصد تولید در هر لیتر از تقطیر | | | | | | | چاه نفت |
|--------------------------------|-------|---------|----------|-----------|------------|-------|---------|
| پلاستیک | مازوت | سوخت جت | نفت منزل | نفت ماشین | روغن ماشین | بنزین | |
| 10 | 13 | 15 | 9 | 10 | 43 | 0W1 | |
| 15 | 16 | 14 | 5 | 12 | 38 | 0W2 | |
| 14 | 12 | 12 | 8 | 8 | 46 | 0W3 | |

جدول ۷.۱۰: تولیدهای تقطیر در هر لیتر نفت خام

| پلاستیک | مازوت | سوخت جت | نفت منزل | نفت ماشین | روغن ماشین | بنزین |
|---------|-------|---------|----------|-----------|------------|-------|
| 0.15 | 0.25 | 0.50 | 0.20 | 0.20 | 0.40 | |

جدوال، خواهیم داشت

$$y(t) = \begin{cases} 1 - e^{-\omega_n t} \left[\cosh \sqrt{\zeta^2 - 1} \omega_n t + \frac{\zeta}{\sqrt{\zeta^2 - 1}} \sinh \left(\sqrt{\zeta^2 - 1} \omega_n t \right) \right] & \zeta > 1 \\ 1 - e^{-\omega_n t} (1 - \omega_n t) & \zeta = 1 \\ 1 - e^{-\omega_n t} \left[\cosh \sqrt{1 - \zeta^2} \omega_n t + \frac{\zeta}{\sqrt{1 - \zeta^2}} \sinh \left(\sqrt{1 - \zeta^2} \omega_n t \right) \right] & \zeta < 1 \end{cases} \quad (18.10)$$

زمان استقرار ^{۱۲} زمانی سپری شدن ($y(t)$) است تا به مقدار نهائی خود در حد متعارفی ده یاشکل ۴.۱۰: زمان نشتی یک مدار RLC برای $\zeta = 1$ ، $\zeta = 0.2$ و $\zeta = 2$ ^{۱۲}settling time

پنج درصد برسد، تعریف می‌شود. شکل (۴.۱۰) پاسخ پله‌واحد را برای سه مقدار ζ بازاء $1 = \omega_n$ نشان می‌دهد.

در این مسئله، می‌خواهیم بهترین مقدار ζ را بنحوی تعیین کنیم که زمان استقرار کمینه شود.

۱. برنامه‌ای به زبان متلب با استفاده از تابع `step_response(zeta, wn, t)` نوشته که پاسخ پله‌واحد $H(s)$ معادله (۱۸.۱۰) بصورت تابعی از ζ و ω_n در فواصل زمانی داده شده محاسبه کند. با رسم پاسخ پله‌واحد بازاء $\zeta = 0.2$ ، $\zeta = 1$ و $\zeta = 2$ جواب خود را صحبت سنجی کنید. فرض کنید $1 = \omega_n$ و فواصل زمانی $0 \leq t \leq 20$ را در قدمهای ۰.۱ ثانیه در نظر گیرید. نتایج شما باید مانند شکل (۴.۱۰) شود.

۲. برنامه دیگری به زبان متلب با استفاده از تابع `find_settling_time(zeta, wn, t, Etol)`

نوشته بطوری که زمان استقرار t_s شکل موج تولید شده با `step_response()` را با تولرانس مشخص $Etol$ بدست آورد. فرض کنید شکل موج $y(t)$ را وقتی اسقراط یافته می‌توان تلقی کرد که اندازه انحراف آن از مقدار نهائی کمتر از تولرانس باشد، یعنی،

$$\left| \frac{y(t) - y(\infty)}{y(\infty)} \right| < E_{tol} \quad (۱۹.۱۰)$$

که در آن $y(\infty)$ مقدار نهائی استقرار یافته $s(t)$ بوده و $Etol$ تولرانس خطای مطلوب است (یعنی پنج یا ده درصد). بنابراین، برای مثال، برای پیدا کردن t_s زمانی که $y(t)$ به پنج درصد مقدار نهائی خود برسد، تابع باید از تکنیک‌های فصل چهارم برای ریشه‌یابی معادله زیر استفاده کند.

$$\left| \frac{y(t) - 1}{1} \right| - 0.05 = 0 \quad (۲۰.۱۰)$$

توجه کنید که در حالت زیرمیرایی، $y(t)$ ممکن است در حوالی تولرانس قبل از استقرار نهائی نوسان کند. بنابراین، باید برای ریشه‌یابی معادله (۲۰.۱۰) از سمت راست شکل موج (یعنی، از $t = \infty$ بجای $t = 0$) جستجو را شروع کنید. در این صورت شما مطمئن خواهید بود که مقدار استقرار نهائی را بعداز بین رفتن نوسانات بدست می‌آورید.

۳. برای بازه $0.2 \leq \zeta \leq 2$ در قدمهای 0.01 ، نمودار t_s را برای تولرانس استقرار 5% رسم کنید. $1 = \omega_n$ را در نظر گیرید. ممکن است بازاء $1 \leq \zeta$ ناپیوستگی در نمودار مشاهده کنید. علت این ناپیوستگی چیست؟

۴. از تابع `fminunc()` متلب استفاده کرده و مقدار ζ ای که زمان استقرار را کمینه می‌کند بدست آورید. این مقدار را با نمودار بخش سوم بالا مقایسه کنید. آیا جواب شما درست بنظر میرسد؟

۵. با استفاده از روش‌های برازش منحنی فصل هشتم، از تابع مرتبه چهارم `polyfit()` متلب استفاده کرده و داده‌های t_s را در مقابل ζ برازش کنید. کمینه کردن روی منحنی برازش شده را دوباره اجرا کنید. آیا جواب شما دقیق‌تر بنظر میرسد؟

فصل ۱۱

سیمیولینک

۱.۱۱ مقدمه

سیمیولینک نام ابزاری در متلب است که برای مدل سازی، شبیه سازی، و تحلیل دینامیکی بکار برده می شود. افراد معمولی از آن برای حل معادلات دیفرانسیلی، مدل سازی سیستمهای فیدبک (پسخور) و پردازش سیگنال ها استفاده می کنند.

با سیمیولینک، مدلها را می توان از روی نوشته یا اضافاتی که میتوان به مدل موجود درست کرد، ساخت. سیمیولینک ها را می توان کنشی واکنشی درست کرد، بدین ترتیب که هر تغییری در پارامترهای را می توان در حالی که سیمیولینک اجرا می کند، انجام داد. سیمیولینک سیستم های خطی و غیرخطی، زمان پیوسته، نمونه برداری شده زمانی، یا ترکیبی از این دو را پشتیبانی می کند.

در سیمیولینک یک واسطه کاربر گرافیکی^۱ برای ساخت مدلها بصورت نمودارهای جعبه ای^۲، با استفاده از عملیات کلیک کردن و کشیدن توسط موس، تدارک دیده شده است. با استفاده از نمایشگرهای مختلف، نتایج شبیه سازی را می توان در طی اجرای سیمیولینک مشاهده نمود. برنامه شامل کتابخانه مفصلی از قطعات ("بلوک ها") است که بوسیله آنها مدل برپا و ساخته می شود. البته، کاربردهای ویژه ای بصورت "جعبه ابزار"^۳ نیز موجود است.

۲.۱۱ تولید یک مدل در سیمیولینک

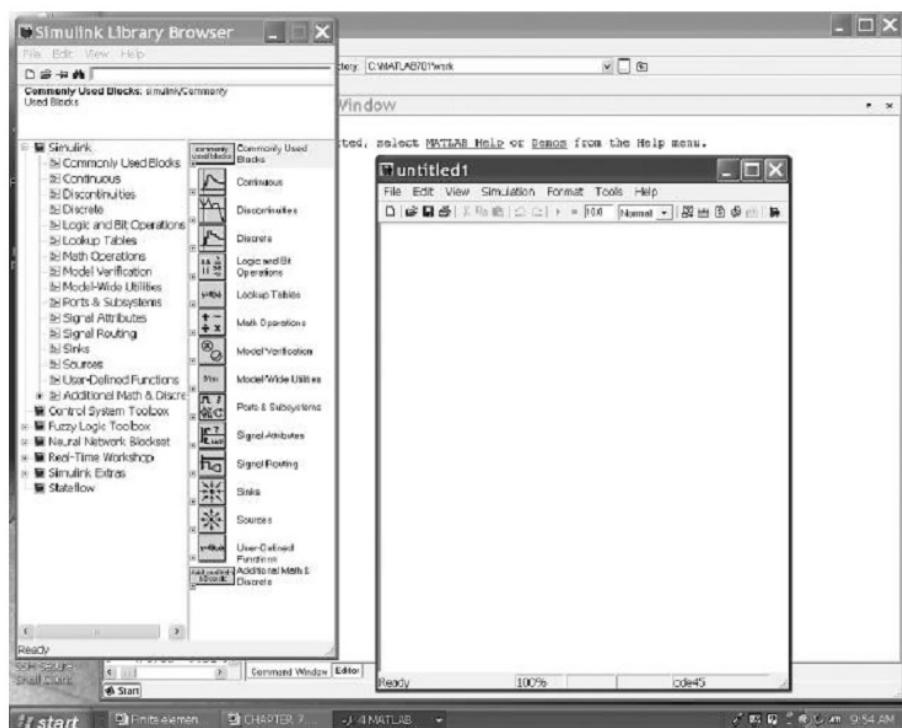
۱. روی آیکن سیمیولینک در فهرست درخواست در پنجره متلب کلیک کنید یا کلمه `simulink` را در پنجره فرمان متلب بنویسید. این برای شما پنجره کتابخانه سیمیولینک را می آورد. به شکل (۱.۱۱) مراجعه کنید.

۲. روی کلمه فایل *File* در پنجره جستجوگر کتابخانه سیمیولینک کلیک کنید. برای مدل جدید، *Open New-Model* یا *New-Model* را برای یک مدل موجود انتخاب کنید. این یک پنجره مدل بدون عنوان (برای حالت مدل جدید) یا پنجره مدل موجود بالا می آورد.

^۱graphical user interface (GUI)

^۲block diagram

^۳toolboxes



شکل ۱.۱۱: جستجوگر کتابخانه سیمیولینک و پنجره مدل بدون عنوان آن

۳. برای تولید یک مدل جدید، لازم است از پنجره جستجوگر کتابخانه بلوک‌هایی را به پنجره مدل جدید کپی کنید. این را می‌توان با روش^۴ کردن^۴ بلوک مخصوص و کشیدن به داخل پنجره جدید انجام داد. برای ساده اتصالات بلوک‌ها، ممکن است لازم باشد که بلوکی را 90° یا 180° بچرخانید. برای انجام این عمل، بلوک را روشن کرده و روی *Format* در نوار منو کلیک کرده، و سپس *Flip Block* را برای (90°) یا *Rotate Block* را برای (180°) انتخاب کنید.

سیمیولینک خیلی رسته‌بندی برای نمایش بلوک‌های کتابخانه دارد؛ آنهایی که برای این فصل مورد نظر است بلوک‌های مورد استفاده عموم^۵، پیوسته‌ها^۶، ناپیوسته‌ها^۷، عملگرهای ریاضی^۸، پایانه‌ها و زیرسamanه‌ها^۹، مسیرگزینی سیگنال‌ها^{۱۰}، متابع^{۱۱}، و توابع تعریف شده^{۱۲} هستند. بلوک‌های عمومی که بارها در این فصل استفاده خواهد شد، ثابت‌ها، ساعت (از کتابخانه منابع)، ضرب، بهره، جمع (از کتابخانه عملیات ریاضی)، انتگرال‌گیری (از کتابخانه پیوسته)، نمایشگر (از کتابخانه سینکSink)، رله (از کتابخانه ناپیوسته‌ها)، کلید، ماکس (از کتابخانه مسیرگزینی سیگنال)، و FCn (از کتابخانه توابع تعریف شده) هستند.

^۴highlighting

^۵Commonly Used Blocks

^۶Continuous

^۷Discontinuities

^۸Math Operations

^۹Ports & Subsystems

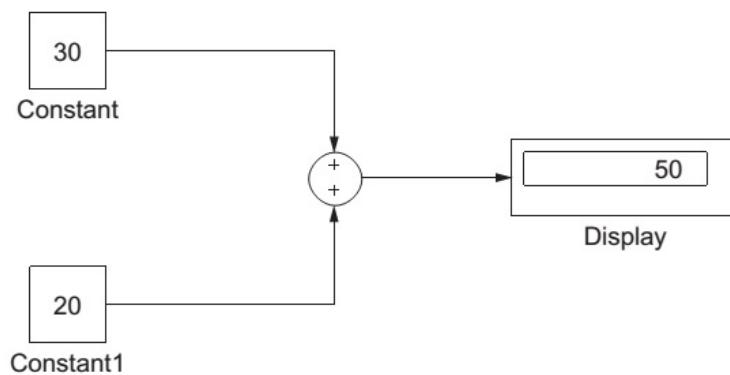
^{۱۰}Signal Routing

^{۱۱}Sources

^{۱۲}User-Defined Functions

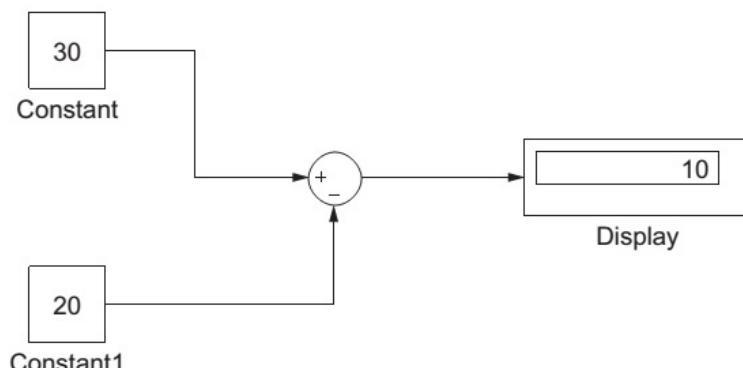
۳.۱۱ نمونه‌ای از بلوک‌های ساختمانی در ساخت یک مدل

۱. جمع دو بلوک ثابت با خروجی نمایش داده شده (به شکل ۲.۱۱ مراجعه کنید). برای قرار دادن مقدار در یک بلوک ثابت، روی بلوک دو بار کلیک کرده و مقدار ثابت را بنویسید. برای اجرای برنامه، روی *Start Simulation* در نوار منو کلیک کرده و روی دکمه (►) در نوار منو کلیک کنید.



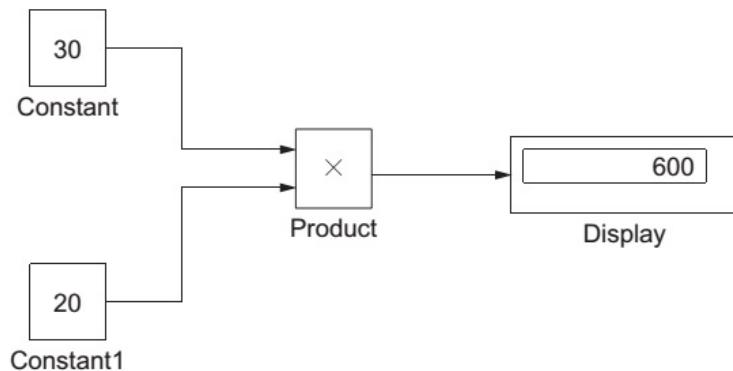
شکل ۲.۱۱: بلوک‌های ثابت، جمع کننده، و نمایشگر برای عمل جمع

۲. تفریق (به شکل ۳.۱۱ مراجعه کنید). برای ساختن، بلوک جمع عمل تفریق را با دو مرتبه کلیک روی بلوک و تغییر علامت آن انجام میدهد.

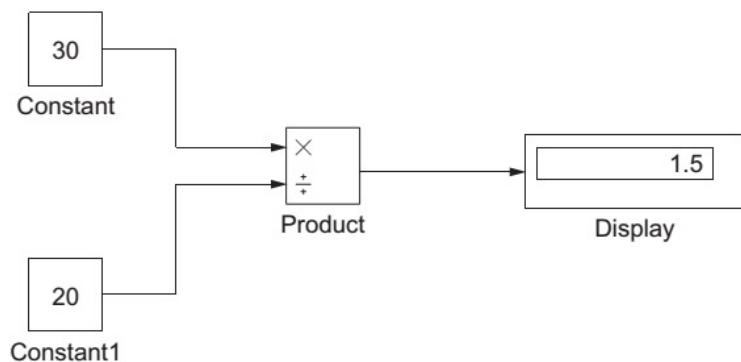


شکل ۳.۱۱: بلوک‌های ثابت، جمع کننده، و نمایشگر برای عمل تفریق.

۳. عمل ضرب دو بلوک را انجام میدهد (به شکل ۴.۱۱ مراجعه کنید).
۴. عمل تقسیم دو بلوک را انجام میدهد (به شکل ۵.۱۱ مراجعه کنید).
۵. در شکل (۶.۱۱) انتگرال‌گیری موج سینوسی نشان داده شده است. پارامترها در بلوک موج سینوسی، فرکانس در دو رادیان برثانیه تنظیم می‌شود. برای مشاهده شکل موج خروجی، ابتدا دکمه اجرا (►) را کلیک کرده و سپس روی بلوک نمایش دوبار کلیک کرده تا نمودار باز



شکل ۴.۱۱: بلوکهای ثابت، ضرب کننده، و نمایشگر برای عمل ضرب.

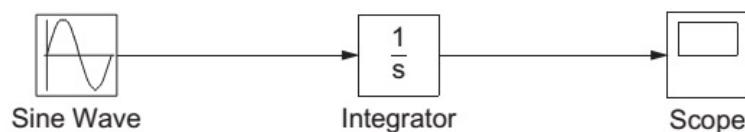


شکل ۵.۱۱: بلوکهای ثابت، ضرب کننده، و نمایشگر برای عمل تقسیم.

شود. سپس روی نمودار کلیک راست کرده و *autoscale* را انتخاب کنید. شکل موج حاصل انتگرال $\int_0^t \sin 2x dx$ خواهد بود:

$$\int_0^t \sin 2x dx = -\frac{1}{2} \cos 2x \Big|_0^t = -\frac{1}{2}(\cos 2t - 1) = \frac{1}{2}(1 - \cos 2t) \quad (1.11)$$

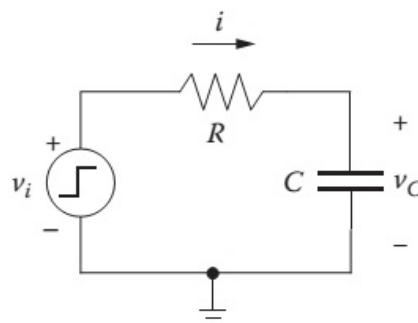
توجه کنید، بعلت این که انتگرال گیر (بطور قراردادی^{۱۳}) دارای شرایط اولیه صفر است، نتیجه از صفر تا یک تغییر خواهد کرد.



شکل ۶.۱۱: بلوکهای موج سینوسی، انتگرال گیر، و نمایش دهنده.

^{۱۳}default

۶. حل یک معادله دیفرانسیل معمولی مرتبه اول ساده. روش حل در مثال ۱.۱۱ با در نظر گرفتن پاسخ پله‌واحد یک مدار RC تشریح شده است.



شکل ۷.۱۱: مدار RC مرتبه اول.

مثال ۱.۱۱:

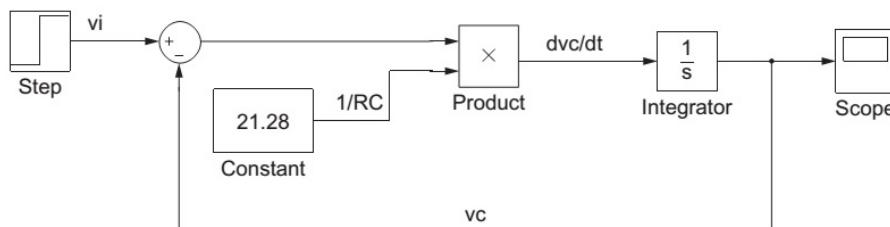
در مدار RC سری شکل (۷.۱۱)، ولتاژ دوسر خازن v_C را نسبت به زمان برای ولتاژ ورودی پله‌واحد آزمایش خواهیم کرد. معادلات حاکم (بطوری که در مثال ۱.۶ آورده شد) بقرار زیر است:

$$\frac{dv_C}{dt} = \frac{1}{RC}(v_i - v_C) \quad (۲.۱۱)$$

که در آن

$$\begin{aligned} R &= 10k\Omega \\ C &= 4.7 \mu F \\ v_i &= \begin{cases} 0 V & t \leq 0 \\ 1 V & t > 0 \end{cases} \quad \text{برای} \quad \begin{cases} t \leq 0 \\ t > 0 \end{cases} \quad \text{برای} \end{aligned}$$

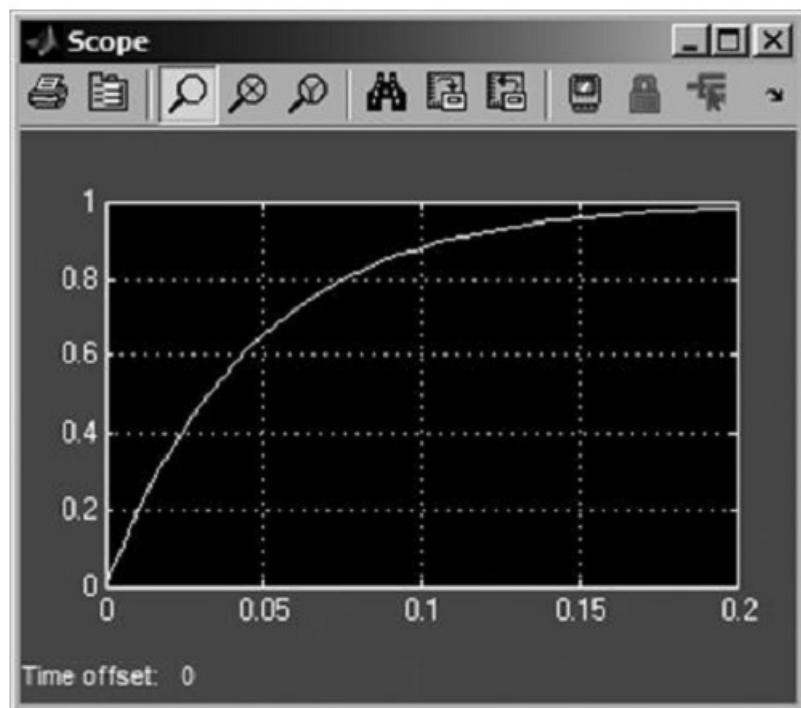
دیاگرام جعبه‌ای برای معادله (۲.۱۱) در شکل (۸.۱۱) در شکل (۹.۱۱) نشان داده شده است.



شکل ۸.۱۱: دیاگرام جعبه‌ای حل معادله دیفرانسیل (۲.۱۱).

۴.۱۱ نکاتی چند در ساختن و اجرای مدل‌ها

۱. برای اتصال خطوط از خروجی یک بلوک به ورودی بلوک دیگر، نشانک موس را روی خروجی بلوک اول قرار داده، کلیک راست موس را انجام داده، و خطی به ورودی بلوک دوم بکشید.



شکل ۹.۱۱: نمایش خروجی مدار RC.

۲. برای اتصال یک نقطه روی یک خط به ورودی یک بلوک، نشانک موس را روی خط قرار داده، کلیک راست موس را انجام داده، خطی به ورودی بلوک بکشید.

۳. برای اضافه کردن اطلاعات حروف و اعداد بالای یک خط، دکمه سمت چپ را بکار برد، دوبار کلیک بالای خط انجام داده، و یک جعبه متنی ظاهر می‌شود. برچسب مطلوب خود را تایپ و جائی دیگر کلیک کرده تا تکمیل شود.

۴. برای مشاهده نتایج روی نمایشگر، دوبار کلیک روی اسکوپ^{۱۴} کرده تا نمودار ظاهر شود. برای انتخاب محور نمودار، کلیک راست روی نمودار کرده و *axis properties* یا *autoscale* کافی است. ضمناً می‌توانید روی آیکن دوربین دو چشمی کلیک کرده تا نمودار بطور خودکار مقیاس بندی شود.

۵. برای تنظیم شرایط اولیه یک انتگرال‌گیر، دوبار کلیک روی بلوک انجام داده و شرایط اولیه را ویرایش کنید.

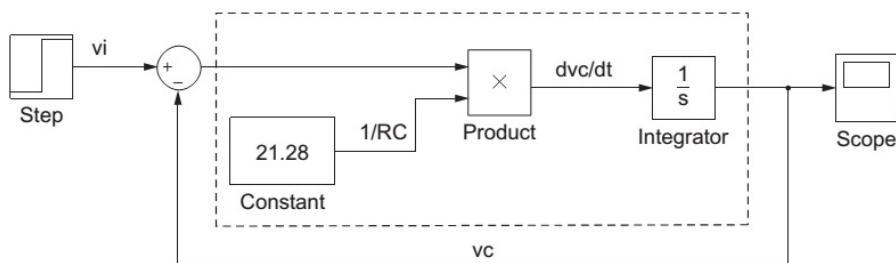
۶. طبق پیش فرض^{۱۵}، سیمیولینک در فاصله زمانی صفر تا ده ثانیه اجرا می‌شود. این زمانها برای خیلی از مدلها نامناسب است (مثل، مدارهای فرکانس بالا). برای ویرایش زمان شروع و خاتمه، در نوار منو، روی *Simulation* کلیک کرده و *Configuration Parameters* را انتخاب کرده، و در جعبه زمانی شروع و خاتمه را ویرایش کنید. راه دیگر، شما می‌توانید زمان خاتمه را در نوار منو تنظیم کنید (زمان شروع پیش فرض صفر است).

^{۱۴}scope^{۱۵}default

۷. برای اجرای سیمیولینک، روی *Simulation* در نوار منو کلیک کرده و *Start* را انتخاب کنید.
راه دیگر، روی دکمه اجرا (▶) کلیک کنید.

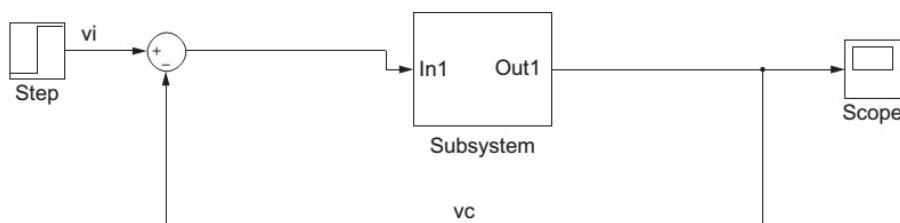
۵.۱۱ ایجاد یک زیرسامانه

فرض کنید ما یک سیستم (سامانه) بزرگ شامل چندین بلوک ساخته، و میخواهیم در ظاهر کل نمودار جعبه‌ای، تعدادی از بلوک‌ها را کاهش دهیم. این را می‌توان با تولید زیرسامانه^{۱۶} انجام داد. زیرسامانه در یک بلوک تکی ظاهر می‌شود. برای تولید زیرسامانه، نشانک (مکان نما)^{۱۷} را در نزدیکی ناحیه‌ای که میخواهیم زیرسامانه ساخته شود قرار دهید و کلیک راست موش را انجام دهید. این باعث می‌شود که یک جعبه نقطه چین تولید شود که می‌توان با کشیدن موس در سراسر ناحیه آنرا بزرگ کرده تا تعدادی از بلوک‌ها را در زیر سامانه شامل شود (شکل ۱۰.۱۱). وقتی دکمه موس را آزاد کنید، منو drop-down ظاهر می‌شود. در اینجا ایجاد زیرسامانه *Create Subsystem* را انتخاب کرده تا چندین بلوک را دیگر یک بلوک زیرسامانه قرار دهد.



شکل ۱۰.۱۱: ایجاد یک زیرسامانه

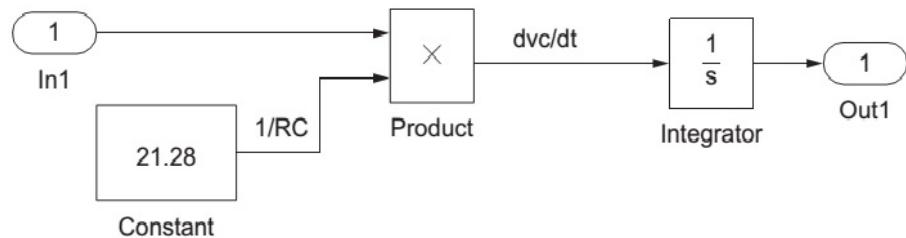
در شکل (۱۰.۱۱)، بلوک‌های ثابت، ضرب، و انTEGRATOR ترکیب شده تا یک زیرسامانه تولید شود. این زیرسامانه ویژه، یک ورودی و یک خروجی دارد، اما بطور کلی، یک زیرسامانه ممکن است چندین ورودی و چندین خروجی داشته باشد. با دوبار کلیک روی زیرسامانه، شما می‌توانید اجزاء داخل آنرا مشاهده کنید. (به شکل ۱۲.۱۱ مراجعه کنید).



شکل ۱۱.۱۱: نمودار جعبه‌ای با یک زیرسامانه برای حل معادله (۲.۱۱).

^{۱۶}Subsystem

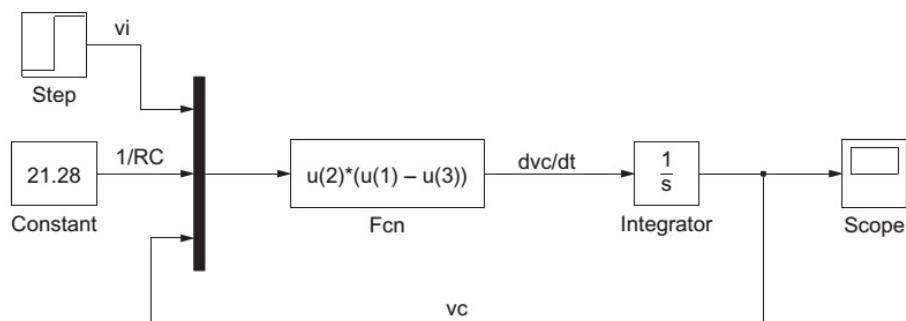
^{۱۷}cursor



شکل ۱۲.۱۱: مدل زیرسامانه

۶.۱۱ استفاده از بلوکهای Fcn و Mux

یک راه دیگر برای دیاگرام جعبه‌ای مثال ۱.۱۱ در شکل (۱۲.۱۱) نشان داده شده است. در این راه حل، بلوکهای Mux و Fcn سیمیولینک برای حل مسئله بکار می‌روند. بلوک Mux بشما اجازه میدهد چندین ورودی (برای تنظیم تعداد ورودی‌ها دوبار کلیک روی بلوک کرده و تعداد ورودی‌ها را ویرایش کنید) را انتخاب کنید. اولین ورودی را با $[1, u]$ و بعدی را با $[2, u]$ و همین طور الی آخر مشخص می‌نمایید. در این مثال، خروجی بلوک Mux باید به ورودی بلوک Fcn رود.



شکل ۱۳.۱۱: نمودار جعبه‌ای برای حل معادله (۲.۱۱) با استفاده از بلوکهای Mux و Fcn

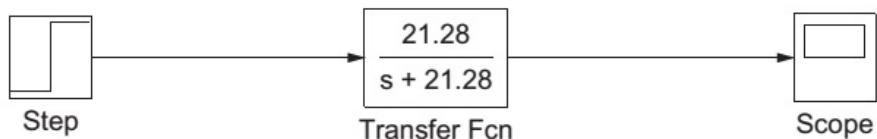
هدف از بلوک Fcn این است که بما اجازه می‌دهد تا توابع متلب و ریاضی دلخواه را در درون مدل تعریف کنیم. در این حالت، رابطه ریاضی در بلوک Fcn بکار رفته تا معادله (۲.۱۱) را بر حسب $[u]$ ها اجرا کند.

۷.۱۱ استفاده از انتقال بلوک Fcn

روش معمول برای حل مسائل مدار این است که امپدانس مختلط را برای خازن و سیم پیچ قرار داده و سپس مدار را مانند مدار مقاومتی حل کرد. برای مدار RC ، امپدانس مقاومت $Z_R = R$ و امپدانس خازن $Z_C = 1/Cs$ (در آن $\omega = j\omega$) است. در این صورت، ولتاژ خازن بسادگی از مقسم ولتاژ مقاومتی بدست می‌آید:

$$v_C = \frac{Z_C}{Z_R + Z_C} v_i = \frac{1/RC}{s + 1/RC} v_i = H(s) v_i$$

که در آن $H(s) = \frac{1/RC}{s + 1/RC}$ می‌نامند. در سیمیولینک، بلوک Fcn بما اجازه میدهد تا تابع تبدیل را مانند شکل (۱۴.۱۱) مدل کنیم.



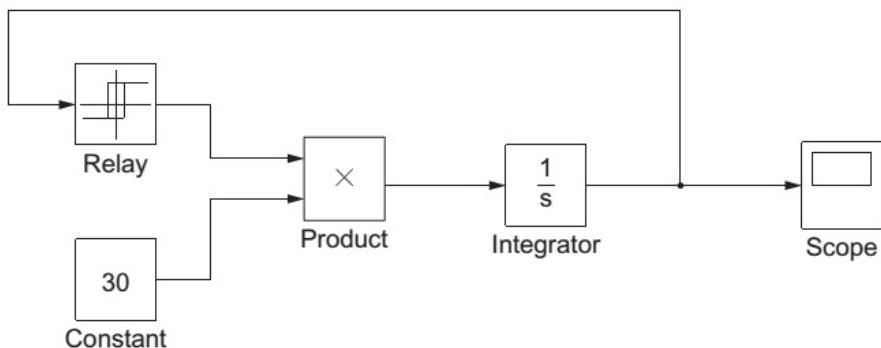
شکل ۱۴.۱۱: نمودار جعبه‌ای با استفاده از بلوک انتقال Fcn.

۸.۱۱ استفاده از بلوک‌های سویچ و رله

رله‌ها و کلیدها، در ادوات کم توان (مانند کنترول کننده‌های الکترونیکی) برای کنترول کردن سیستمهای توان بالا (مثل کوره‌ها و دیکبخار) بکار می‌روند. سیمیولینک دارای بلوک‌های سویچ و رله است که می‌توان برای شبیه‌سازی اینگونه سیستم‌ها بکار برد.

مثال ۲.۱۱:

در یک سامانه کرمایشی منزل، حسگر دما برای روشن و خاموش کردن دیکبخار بکار می‌رود تا حرارت منزل در دمای متبوعی قرار گیرد. البته، چون اکثر دیک‌های بخار در مرتبه بیشمار نمی‌توانند خاموش و روشن شوند (یعنی بعداز روشن چند دقیقه طول می‌کشد تا گرم شود و بعداز خاموش کردن چند دقیقه طول می‌کشد که سرد شود)، کنترول دمای اطاق مستلزم قدری پس‌ماند^{۱۹} است تا از قطع و وصل متناوب دیکبخار (که باعث پرباری دیک می‌شود) جلوگیری کند. این مفهوم را می‌توان با معادله دیفرانسیل زیر که دمای T بین تغییر دو نرخ ثابت 20°C و 22°C تنظیم شود، نمایش داد:



شکل ۱۵.۱۱: نمودار جعبه‌ای با استفاده از بلوک رله از مثال ۲.۱۱

$$\frac{dT}{dt} = c \quad \text{که در آن} \quad c = \begin{cases} +30 & \text{اگر } T \leq 20 \\ -30 & \text{اگر } T \geq 22 \end{cases}$$

^{۱۸}transfer function

^{۱۹}hysteresis

نمودار جعبه‌ای این سامانه شامل یک انتگرال‌گیر، یک مقدار ثابت، یک رله، یک ضرب کننده، و یک نمایش دهنده (شکل ۱۶.۱۱) است. رله برای تغییر علامت c بکار می‌رود و بستگی به این دارد که رله روشن یا خاموش باشد. پارامترهای رله را می‌توان با دوبار کلیک روی بلوک رله و برایش کرده و پارامترها را بمقادیر زیر تنظیم کرد:

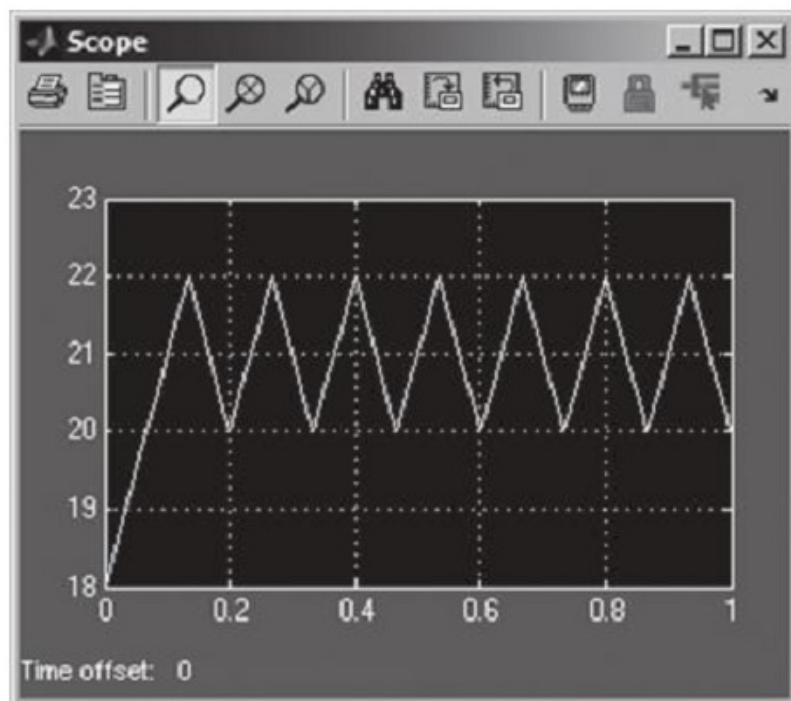
- نقطه روشن سویچ = 22

- نقطه خاموش سویچ = 20

- خروجی وقتی روشن است = -1

- خروجی وقتی خاموش است = +1

فرض می‌کنیم که دمای اولیه اطاق $T = 18^\circ C$ است، و این مقدار برای پارامترهای بلوک انتگرال‌گیری وارد می‌شود. در شروع شبیه‌سازی، $T \leq 20$ بنا براین، رله قطع بوده و خروجی رله +1 و باعث می‌شود که T افزایش یابد. خروجی رله روی همان +1 باقی می‌ماند تا این که T به 22 برسد. در این موقع رله روشن شده و خروجی آن -1 می‌شود. این باعث می‌شود که T کاهش یابد. خروجی رله در -1 باقی مانده تا T به 20 برسد و رله قطع کند و خروجیش دوباره به +1 برگردد. این مراحل ادامه یافته تا زمان شبیه‌سازی به آخر برسد. خروجی نمایشگر در شکل (۱۶.۱۱) نشان داده شده است.



شکل ۱۶.۱۱: خروجی نمایشگر مثال ۲.۱۱

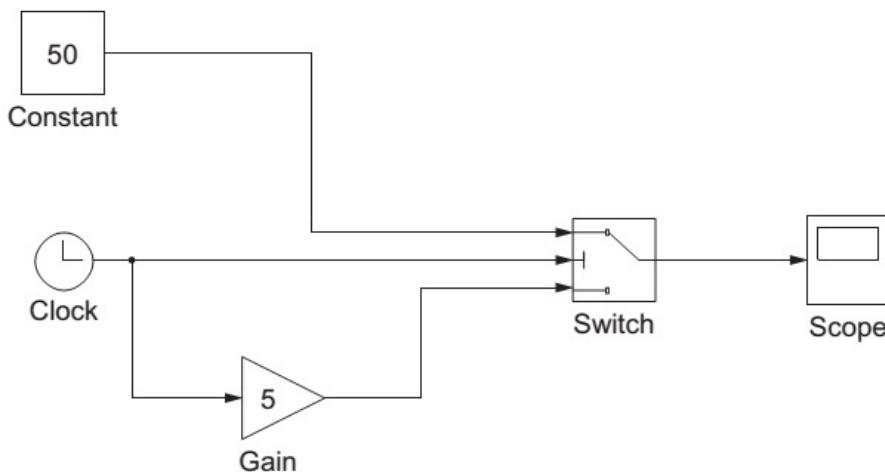
مثال ۳.۱۱:

برخی مسائل شامل تابعی هستند که در زمان $t_1 \leq t \leq t_2$ تغییر کرده و برای زمان $0 \leq t \leq t_1$ ثابت است. این نوع تابع را می‌توان با بلوک سویچ که یک سویچ DPT^{۲۰} می‌سازد مدل کرد. این سویچ یک ترمینال اضافی دارد که کلید را باز یا بسته می‌کند.

فرض کنید

$$y = \begin{cases} 5t & \text{برای } 0 \leq t \leq 10 \\ 50t & \text{برای } 10 < t \leq 20 \end{cases}$$

مدل سیمیولینک برای این مسئله در شکل (۱۷.۱۱) نشان داده شده است. پارامترها برای این سویچ بقرار زیر است (توجه کنید که ترمینال u_1 ورودی بالائی، u_2 ورودی وسطی [کنترول کننده]، و u_3 ورودی پائینی سویچ است):



شکل ۱۷.۱۱: نمودار جعبه‌ای با استفاده از بلوک سویچ از مثال ۳.۱۱

$$\begin{aligned} \text{معیار برای عبور ورودی اول:} \\ u_2 \geq \text{آستانه} \end{aligned}$$

$$\text{آستانه} : 10$$

خروجی حاصل در شکل (۱۸.۱۱) نشان داده شده است. توجه کنید که ما از بلوک ساعت برای تعیین u_3 بمنظور گذشت زمان استفاده کردیم.

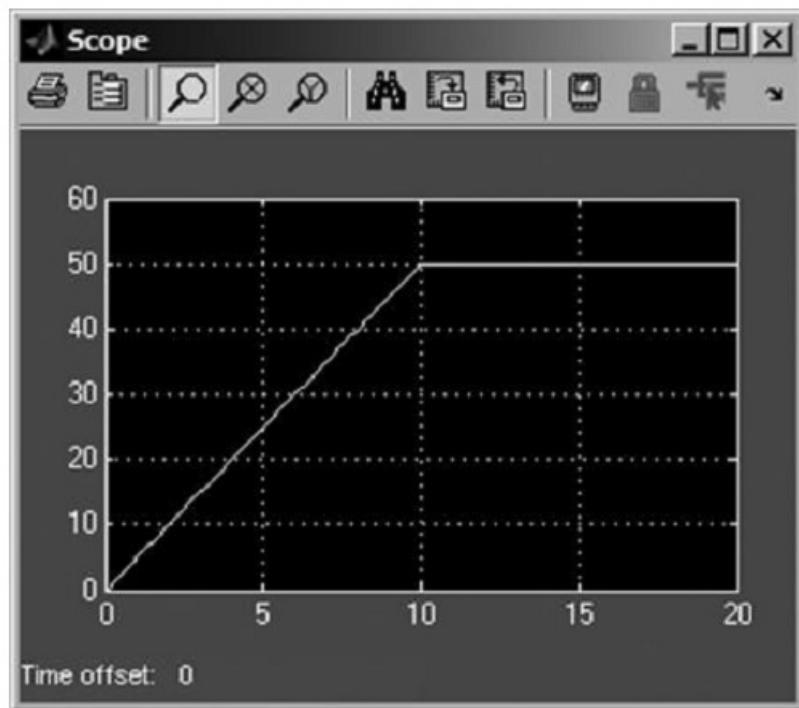
۹.۱۱ بلوک‌های توابع مثلثاتی

توابعی از قبیل سینوس، کسینوس، و تانژانت را می‌توان از طریق بلوک تابع مثلثاتی^{۲۱} که در کتابخانه عملیات ریاضی سیمیولینک قرار دارد بدست آورد. ورودی این بلوک آرگومان تابع مثلثاتی است. اگر آرگومان شامل متغیر مستقل t (مانند $\sin \omega t$) باشد، در این صورت می‌توانیم از بلوک ساعت^{۲۲} برای

^{۲۰}double-pole, single-throw

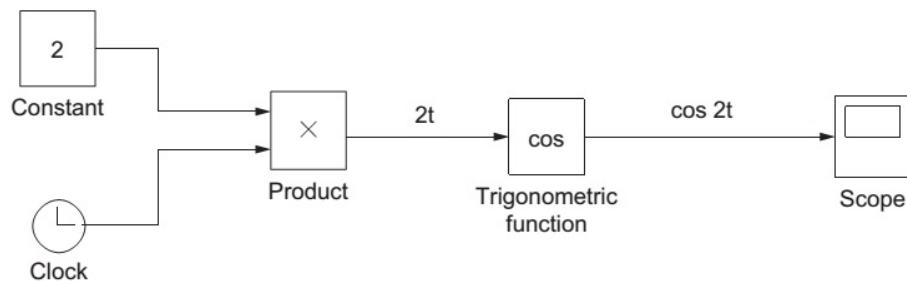
^{۲۱}Trigonometric Function

^{۲۲}Clock block



شکل ۱۸.۱۱: خروجی نمایشگر مثال ۳.۱۱

مقدار t استفاده کنیم. این در شکل (۱۹.۱۱) جایی که ما مقدار $\cos 2t$ محاسبه می‌کنیم، نشان داده شده است.



شکل ۱۹.۱۱: نمودار جعبه‌ای با استفاده از بلوک‌های ساعت، ضرب‌کننده، و توابع مثلثاتی.

مثال ۴.۱۱: شبیه‌سازی مدار RLC

در بخش ۱۷.۲، مدار RLC موازی را مطالعه کردیم. اگر در شکل (۱۴.۲) یک جریان ورودی سینوسوئیدال استفاده کرده و فرض کنیم که کلید در لحظه $t = 0$ بسته شود، رابطه حاکم خواهد بود

$$\frac{d^2 i_L}{dt^2} + \frac{1}{RC} \frac{di_L}{dt} + \frac{1}{LC} i_L = \frac{1}{LC} I_0 \sin \omega t \quad (۳.۱۱)$$

که در آن مدار با یک جریان سینوزوئیدال بفرکانس ω و اندازه I_0 "تغذیه" شده است. مدل سیمیولینک در شکل (۲۰.۱۱) این معادله دیفرانسیل درجه دوم را برای مقادیر و شرایط زیر حل می‌کند:

$$\begin{aligned} C &= 1 \mu F \\ L &= 10 mH \\ I_0 &= 5 mA \\ \omega &= 2000 rad/s \\ i_L(0) &= 2 mA \\ i_L(0) &= 0 \end{aligned}$$

خروجی این شبیه‌سازی در شکل (۲۱.۱۱) نشان داده شده است. شبیه‌سازی و پارامترهای فضای کار^{۲۳} برای این مثال بقرار زیر است:

- زمان شبیه‌سازی:
زمان شروع: 0.0
زمان خاتمه: 0.015

- اختیارات حل کننده:

نوع: پله متغیر
 حل کننده: ode45(Dormand-Prince)
 اندازه قدم بیشینه: خودکار auto
 اندازه قدم کمینه: خودکار auto
 اندازه قدم اولیه: خودکار auto
 تولرانس نسبی: $1e - 3$
 تولرانس مطلق: خودکار auto

- پارامترهای فضای کار:

نام متغیر: i
 حد نقاط داده: inf
 اعشار: 1
 زمان نمونه: -1
 فرمت ذخیره: آرایه

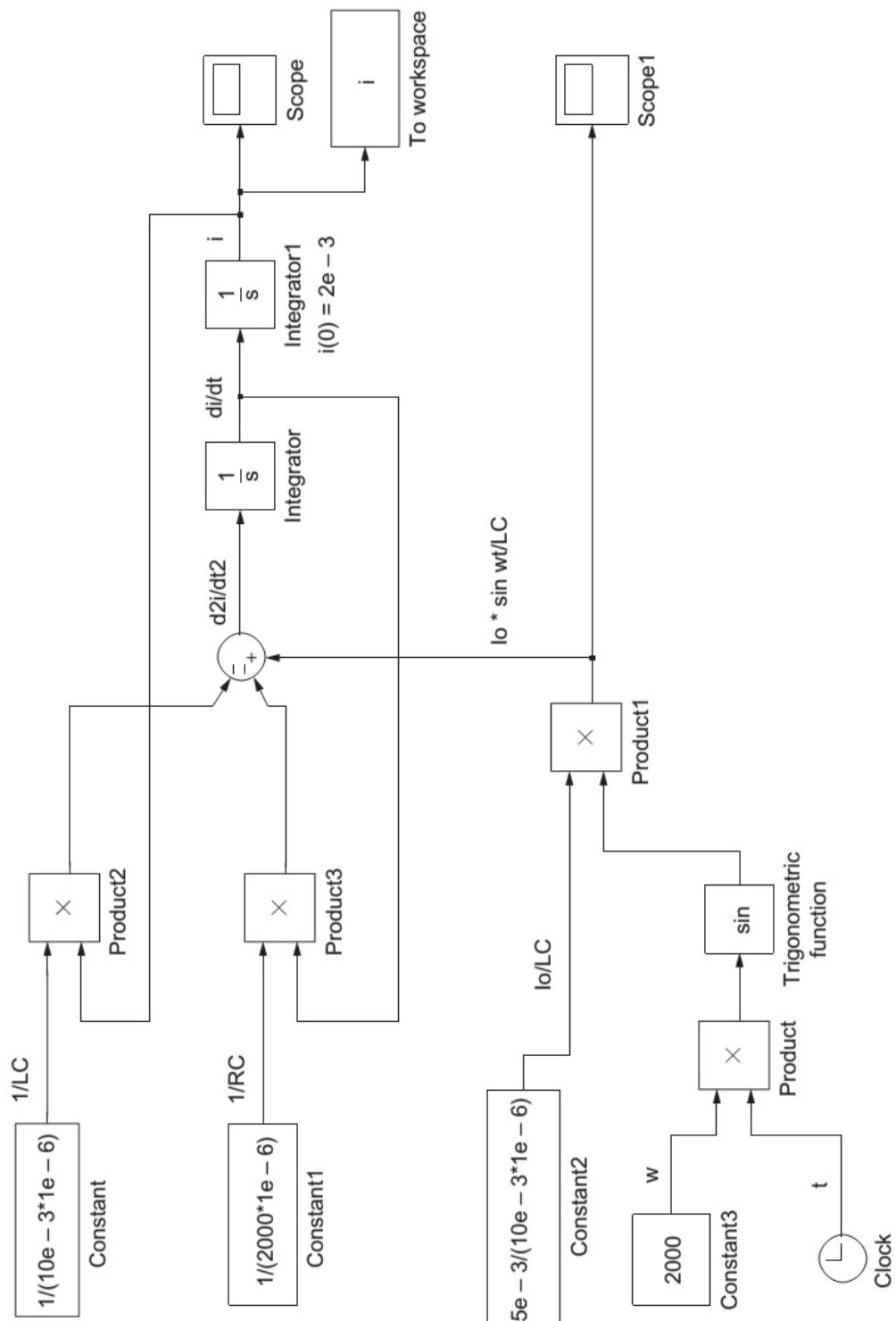
توجه کنید که شما می‌توانید مقادیر خروجی سیمیولینک را برای عملیات ریاضی یا چاپ به داخل متغیرهای متلب صادر کنید. این را می‌توانید با بلوک فضای کار مطابق شکل (۲۰.۱۱) انجام دهید. بعداز اجرای سیمیولینک، آن متغیرهایی که برای متلب نیاز دارید موجود است. وقتی از فضای کار استفاده می‌کنید، مطمئن شوید که Save Format property روی آرایه Array تنظیم شده است.

مسائل

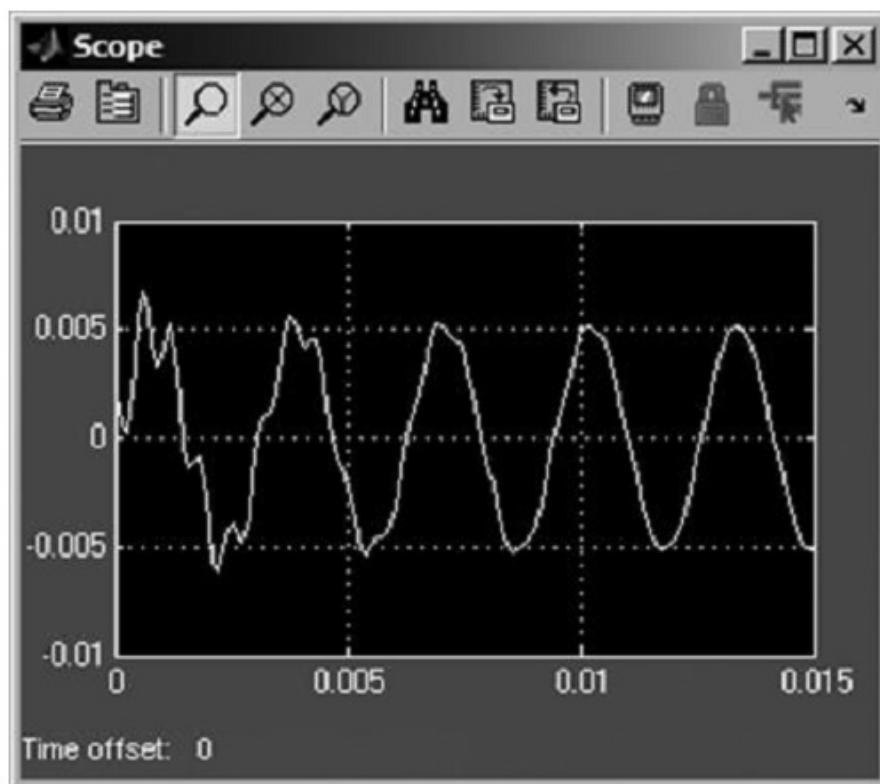
۱. اینتابع را با سیمیولینک بسازید:

$$y = 5t^2 + 4t + 2$$

- ۱- از بلوک ساعت، سه مقدار ثابت، دو ضرب کننده، و یک جمع کننده استفاده کنید.



شکل ۲۰.۱۱: نمودار جعبه‌ای برای حل معادله درجه دوم رابطه (۳.۱۱).



شکل ۲۱.۱۱: خروجی نمایشگر برای مثال ۴.۱۱

- ۲- از بلوک ساعت، سه مقدار ثابت، یک بلوک Mux ، یک بلوک Fcn ، یک بلوک نمایشگر، و دو بلوک فضای کار استفاده کنید. بلوک نمایشگر باید نمودار y را بر حسب t نمایش دهد. ضمناً، برنامه‌ای بزبان متلب نوشته که جدول y را در مقابل t در پنجره فرمان بسازد.
- ۳- از یک مقدار ثابت و دو انتگرال گیر استفاده کنید.

پروژه‌ها

- پروژه ۱.۱۰

مراجع

- [1] Kernighan, B.W., and Ritchie, D.M., *The C Programming Language*, 2nd ed., PrenticeHall, Englewood Cliffs, NJ, 1988.
- [2] Sze, S.M., *Physics of Semiconductor Devices*, 2nd ed., Wiley, New York, 1981.
- [3] Churchill, R.V., *Operational Mathematics*, 2nd ed., McGraw-Hill, New York, 1958.
- [4] Kreyszig, E. *Advanced Engineering Mathematics*, 8th ed., Wiley, New York, 1999.
- [5] A.S. Sedra and K.C. Smith, *Microelectronic Circuits*, 3rd ed., Saunders College, Philadelphia, 1989.
- [6] 2N3904/MMBT3904/PZT3904 NPN general purpose amplifier, Fairchild Semiconductor data sheet, <http://www.fairchildsemi.com>, 2011.
- [7] Siebert, W.M., *Circuits, Signals, and Systems*, MIT Press, Cambridge, MA, 1986.
- [8] Cooley, J.W., and Tukey, J.W., An algorithm for the machine calculation of complex Fourier series, *Mathematics of Computation*, 19:297–301, 1965.
- [9] Oppenheim, A.V., and Schafer, R.W., *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [10] Multifrequency push-button signal reception, ITU-T Recommendation Q.24, <http://www.itu.int>.
- [11] Wylie, C.R., *Advanced Engineering Mathematics*, McGraw Hill, New York, p. 596.
- [12] McNamara, J., *Technical Aspects of Data Communication*, 3rd ed., Butterworth Heinemann, Oxford, UK, 1988.
- [13] Johnson, D.E., Hilburn, J.L., and Johnson, J.R., *Basic Electric Circuit Analysis*, 4th ed., Prentice Hall, 1990.
- [14] Knuth, D.E., *The TeXbook*, Addison Wesley, Boston, 1984. Kreyszig, E., Advanced Engineering Mathematics, 8th ed., Wiley, New York, 1999.

نمايه

- جی اس ام، ۲۲۵
- حالت ماندگار، ۲۱۳
- حساب دیفرانسیل و انتگرال، ۲۷۵
- حفرها، ۱۸۷
- خاصیت دوگانگی، ۲۳۵
- خط نقطه‌گذار مکعبی، ۲۶۶
- خطای دستوری، ۶۹
- خطای مریع میانگین، ۲۵۹
- خطای مرخم محلی، ۱۶۳
- خود راه انداز، ۷
- دترمینان، ۸۹
- درشت، ۱۲
- درصد خطای جهانی، ۱۶۳
- درون‌بایی، ۶۰
- درین-چاہک، ۱۲۴
- دسترسی تقسیم کد (CDMA)، ۲۲۵
- رانج کوتا، ۱۵۵
- روش اولر، ۱۵۵
- روش تفاضلات محدود، ۱۷۴
- روش تندترین نزول، ۲۷۶
- روش سیمپسون، ۹
- روش ضمنی، ۱۵۸
- روش نیوتون-رافسون، ۱۱۵
- رکرسیون، ۲۶۱
- زمان استقرار، ۱۲۳
- زمان نمونهبرداری، ۲۳۸
- زیربرنامه، ۴۲
- زیرسامانه، ۳۰۷
- زیرمیرا، ۷۴
- سالان-کی، ۱۸۵
- سر، ۲۴۴
- سری تیلور، ۱۳۱
- سری تیلور مرخم، ۱۶۳
- سری فوریه، ۲۶۹
- سطر پاشنه، ۹۶
- سنجرش از دور-تله‌متري، ۲۲۵
- آلگوريتم، ۹
- ابعاد، ۸۹
- اشکال زدائی، ۱۱
- اعوجاج هارمونیکی کل، ۸۲
- باترورث، ۲۳۷
- بالک، ۱۲۵
- باند عبور، ۲۳۶
- براژش منحنی، ۶۶
- بسط بینم، ۷۶
- بسط سری تیلور، ۳۸
- بلوک ساعت، ۳۱۱
- تابع انتقال - تابع تبدیل، ۳۰۹
- تابع تقریبی، ۲۵۷
- تابع دستی، ۴۷
- تابع دلتا، ۱۹۸
- تابع دلتای کرونکر، ۲۴۲
- تابع هدف، ۲۷۵
- تابع واحد نمونه، ۲۴۲
- تابع پنجره، ۲۳۷
- تبدیل فوریه، ۲۲۵
- تبدیل فوریه سریع، ۲۴۰
- تبدیل فوریه گسسته، ۲۳۹
- تبدیل لایپلاس، ۱۹۵
- تحرک الکترونها، ۱۲۵
- تداخل علائم، ۲۲۵
- تراشه، ۱۲۷
- ترانهاد، ۸۸
- تقریب میلر، ۲۱۸
- تکین، ۹۸
- ثابت بولتزمن، ۲۷۱
- ثابت زمانی، ۱۵۹
- جبه بول، ۱۲۶
- جهبه ابزار، ۱۰
- جهبه ابزار، ۳۰۱
- جواب خصوصی، ۸۰
- جواب همگن، ۸۰

- نفوذپذیری مغناطیسی، ۱۵۲
 نمودار بود، ۲۱۳
 نمودار جریان، ۱۰
 نمودار جعبه‌ای، ۳۰۱
 نمودار میخی، ۵۳
 نمونه اولیه، ۱۱
 نمونهبرداری، ۲۳۷
 نویز حاشه‌ای، ۱۲۷
 واحد پردازش مرکزی، ۳۵
 واسط کاربر گرافیکی، ۳۰۱
 واگذاری، ۳۵
 ولتاژ آستانه، ۱۲۵
 ولتاژ حرارتی، ۲۷۱
 پاسخ ضربه‌ای، ۲۱۵
 پاسخ ضربه‌ای محدود، ۲۴۴
 پاسخ فرکانسی، ۲۱۳
 پاول دیراک، ۱۹۸
 پردازش سیگنالهای دیزیتالی، ۲۳۷
 پس‌ماند، ۳۰۹
 پله‌واحد، ۲۹۷
 پنجره بارلت، ۲۵۰
 پنجره هامینگ، ۲۵۰
 پیش‌فرض، ۳۰۶
 چه‌بی‌چف، ۲۳۷
 گرمگیر، ۱۹۰
 گیت-دوازه، ۱۲۴
 سه-قطري، ۱۷۵
 سیگنال غيرتناوبي، ۲۳۱
 صريح، ۱۵۷
 صفحه گستردگ، ۶۹
 ضرب داخلی، ۸۹
 ضرب کننده‌های لاغرانژ، ۲۸۴
 ضريب ايده‌آل، ۲۷۳
 ضريب ميرائي، ۲۹۷، ۷۹
 طراحی بكمک كامپيوتر، ۷
 طول قدم، ۱۵۶
 عناصر تاخير دهنده، ۲۴۲
 عنصر پاشنه، ۹۶
 فرآيند دوبخشی، ۱۱۴
 فراميرا، ۷۴
 فركانس طبيعی، ۷۹
 فيلتر بالاگذر، ۲۳۶
 فيلتر ميان نگذر، ۲۳۶
 فيلتر ميان گذر، ۲۳۶
 فيلتر پائين گذر، ۲۳۶
 قابل تحقق، ۲۳۷
 قابلیت تحرک، ۶۵
 قاعده کرامر، ۲۵۸، ۹۵
 قانون سيمپسون، ۱۳۱
 ماتريس سه-قطري، ۱۷۵
 ماتريس همانی، ۸۹
 متعادل کننده، ۲۳۶
 متعامد، ۲۲۶
 متغيرهای کنترول، ۲۷۵
 متمم، ۱۲۷
 محدوديت، ۲۷۵
 معادلات ديفرانسيل معمولي، ۱۵۵
 معادلات متعالي، ۱۱۳، ۱۳
 معکوس کننده، ۱۲۶
 مقدار تصحيح، ۱۵۸
 مقدار پيشگوئي، ۱۵۸
 منبع، ۱۲۴
 مكان‌نما، ۲۹
 ميانگين وزن‌دار، ۱۶۴
 ميرائي بحراني، ۷۴
 ناحيه تخلية، ۱۸۸
 نرخ نمونهبرداری، ۲۳۸
 نشانک، ۳۰۷
 نظريه باقیماندها، ۱۳۴