پردازش سیگنال‌های بیولوژیکی

سرشناسه: مولایی، خدیجه، ۱۳۶۷-
عنوان و نام پدیدآور: پردازش سیگنال‌های بیولوژیکی/ مؤلفان خدیجه مولایی، حسین بلوچیان و مریم احمدی
مشخصات نشر: تهران، ساکو ۱۳۹۶.
مشخصات ظاهری: ۲۴۵ص.: مصور، نمودار(بخشی رنگی)
شابک: ۱-۹۸-۸۰۶۲-۶۰۰-۹۷۸
وضعیت فهرست‌نویسی: فیپا
یادداشت: کتابنامه: ص. [۲۴۴]-۲۴۵.
موضوع: متلب
موضوع: پردازش سیگنال‌ها
موضوع: پردازش سیگنال‌ها ـ شیوه‌های رقمی
شناسه افزوده: بلوچیان، حسین، ۱۳۵۸-
شناسه افزوده: احمدی، مریم، ۱۳۴۵
رده‌بندی کنگره: ۱۳۹۶ ۴پ۹م/۹/۵۱۰۲TK
رده‌بندی دیویی: ۶۲۱/۳۸۲۲
شماره کتابشناسی ملی: ۴۷۹۱۹۰۸

پردازش سیگنال‌های بیولوژیکی

مؤلفان:
مهندس خدیجه مولایی، مهندس حسین بلوچیان
و دکتر مریم احمدی

انتشارات ماهکو

تهران ۱۳۹۶

پردازش سیگنال‌های بیولوژیکی
خدیجه مولایی، حسین بلوچیان و مریم احمدی
واژه‌نگار: انتشارات ساکو
طرح جلد: انتشارات ساکو
لیتوگرافی: طاووس رایانه، چاپخانه: عطا
سال نشر: ۱۳۹۶
چاپ اول، شمارگان: ۱۰۰۰
قیمت: ۲۵۰,۰۰۰ ریال

* * *

حق چاپ محفوظ است.

تلفن مرکز پخش : ۶۶۴۹۴۵۲۴-۶۶۵۷۲۶۹۲-۰۲۱
www.nashresaco.ir

تقدیم به

ام المصائب کربلا حضرت زینب (س)

فهرست مطالب

عنوان صفحه

پیشگفتار ۱۰

فصل اول: آموزش نرم‌افزار متلب

معرفی نرم‌افزار متلب ۱۲
M-FILE ۱۴
Script M-file ۱۵
Echo کردن دستورات ۱۶
ساختار Script M-file ۱۶
تابع M-file ۱۷
اضافه کردن توضیحات ۱۹
استفاده از دستور DEMO ۱۹
پنجره HELP در متلب ۱۹
باز کردن پنجره Help ۲۰
نحوه دیدن صفحات Help با دستور Help ۲۰
نمایش صفحه help یک دستور با دستور doc ۲۱
انواع متغیرها و مقداردهی به متغیرها در متلب ۲۱
متغیرهای عددی ۲۲
متغیرهای رشته‌ای ۲۲
متغیرهای سمبلیک ۲۳
نمایش مقادیر متغیرها در داخل یک عبارت fprintf . ۲۳
معرفی چند دستور ساده در متلب ۲۳
دستور syms ۲۳
دستور expand ۲۴
دستور factor ۲۴
دستور simplify ۲۵
دستور simple ۲۵
دستور syms ۲۵
دستور whos ۲۵
دستور clear ۲۶
دستور clc ۲۶
عملگرهای ریاضی ۲۷
عملگرهای ریاضی مشخص شده با نماد ۲۷

عنوان صفحه

عملگرهای ریاضی مشخص شده با دستور ۲۸
نمادهای پرکاربرد در متلب ۲۸
دستور FORMAT ۲۹
ایجاد نقاط با فواصل خطی دستور LINSPACE ۳۰
ایجاد نقاط با فواصل لگاریتمی ۳۱
محاسبه حد در متلب ۳۱
دستور limit ۳۱
محاسبه حد راست یا حد چپ ۳۲
محاسبه حد بی نهایت ۳۲
انتگرال‌گیری در متلب ۳۲
انتگرال نامعین ۳۳
انتگرال معین ۳۳
محاسبه انتگرال برای عبارت‌هایی که نمی‌توان به صورت سمبلیک انتگرال آنها را محاسبه نمود ۳۳
محاسبه انتگرال‌های چندگانه ۳۴
توابع مخصوص گرد کردن اعداد در متلب ۳۴
دستور round ۳۵
دستور fix ۳۵
دستور ceil ۳۵
دستور floor ۳۵
بردارها و ماتریس‌ها در متلب ۳۶
بردارها در متلب ۳۶
بردارها با عناصر منظم ۳۶
ماتریس‌ها در متلب ۳۸
نحوه تعریف ماتریس‌ها در متلب ۳۸
توابع ماتریسی در متلب ۳۹
انجام عملیات ریاضی بر روی ماتریس‌ها ۴۲
چندجمله‌ای‌ها ۵۰
ریشه‌های یک چندجمله‌ای ۵۰
دستور roots ۵۱
دستور poly ۵۱
محاسبه مقدار یک چندجمله‌ای ۵۱

دستور Polyder ۵۱
دستور polyint ۵۲
دستور conv ۵۲
دستور deconv ۵۲
مشتق چند جمله ای ۵۲
برازش منحنی چندجمله‌ای ۵۳
حل دستگاه معادلات در متلب ۵۴
مشتق‌گیری در متلب ۵۴
مشتق‌گیری از عبارت‌های سمبلیک ۵۴
مشتق مرتبه دوم و مرتبه‌های بالاتر ۵۵
مشتق گرفتن از یک تابع چند متغیره، بر حسب یک متغیر خاص ۵۶
حل معادله دیفرانسیلی در متلب با دستور DESOLVE . ۵۶
حل معادلات دیفرانسیل مرتبه دوم یا بالاتر ۵۷
حلقه‌های تکرار ۵۷
حلقه for ۵۷
اجرای دستور if ۵۸
ساختار تکرار با دستور WHILE ۶۱
دستور SWITCH CASE ۶۲
ترسیم گرافیکی توابع ۶۴
ترسیم شکل در متلب با دستور plot ۶۴
اضافه کردن عنوان ۶۵
اضافه کردن برچسب به محور ۶۵
اضافه کردن اختصارات ۶۶
برای تعیین رنگ پس زمینه نمودارها ۶۷
ترسیم گرافیکی توابع ۶۸
ترسیم ماتریس‌ها در متلب به صورت یک تصویر ۶۹
نمایش خطوط با رنگ‌ها و علامت‌های مختلف ۷۲
نمایش پس زمینه شکل به صورت چهارخانه ۷۵
نحوه استفاده از دستور figure ۷۶
رسم چند شکل کنار هم در یک پنجره ۷۶
دستور plot3 ..... ۷۸
رسم سطوح به صورت سه بعدی ۷۹
ایجاد تصاویر متحرک ۸۰
ایجاد شکل موج دندان اره‌ای و موج مربع ۸۰
شکل موج نامتناوب ۸۱
ایجاد پالس گاوسی مدوله سینوسی و محاسبه مسیر ورودی یا ماتریس ۸۲
ایجاد نوسان سیگنال در یک فرکانس تعیین شده توسط بردار ورودی ۸۳
قطار پالس ۸۴
برخی دستورات دیگر در رسم نمودار ۸۵
محاسبه تبدیل فوریه ۸۶
تبدیل فوریه گسسته ۸۶
معکوس تبدیل فوریه ۸۶
تبدیل فوریه سریع ۸۷
آنالیز فوریه پایه ۸۷

فصل دوم: آشنایی با سیگنال‌های بیولوژیکی

مقدمه ۹۱
سیستم تجهیزات پزشکی پایه ۹۴
کاربرد تحلیل سیگنال‌های بیولوژیکی ۹۸
نمونه‌هایی از سیگنال‌های بیولوژیکی رایج ۱۰۰
نوار قلب ۱۰۱
نوار مغز ۱۰۳
الکترومایوگرام ۱۰۹
فونوکاردیوگرام ۱۱۱
دیگر سیگنال‌های بیولوژیکی ۱۱۲
مراحل پردازش سیگنال‌های زیستی ۱۱۳
کسب سیگنال‌های زیستی ۱۱۳
دیجیتالی شدن سیگنال‌های زیستی: نمونه‌برداری و تعیین ۱۱۴
نویز ۱۱۵
دقت و صحت ۱۱۵
تجزیه و تحلیل انتزاعی ۱۱۵

فصل سوم: سیگنال‌های زمان گسسته و سیستم‌ها

سیگنال‌های زمان گسسته و سیستم‌ها ۱۱۸
سیگنال زمان گسسته ۱۱۹

نمونه‌برداری ۱۱۹
آلیاسینگ ۱۲۰
دنباله‌ها ۱۲۱
عملیات پایه برای سیستم‌های زمان گسسته ۱۲۲
تولید (مدولاسیون) ۱۲۲
افزودن ۱۲۲
ضرب ۱۲۳
زمان معکوس (پدیده تا شدن) ۱۲۳
انشعاب ۱۲۳
تغییر زمانی ۱۲۴
مقیاس‌گذاری زمان ۱۲۴
ترکیبی از عملیات ۱۲۵
نمونه‌هایی از عملیات دنباله ۱۲۶

فصل چهارم: تبدیل فوریه

تبدیل فوریه ۱۳۰
فرکانس گسسته ۱۳۲
تبدیل فوریه سریع ۱۳۳
تبدیل فوریه گسسته ۱۳۵
محاسبات DFT با استفاده از ماتریس رابطه ۱۳۹
اثر PICKET FENCE ۱۴۰
اثر برش ۱۴۱
نمونه‌های استفاده از DFT برای محاسبه طیف مقدار ۱۴۳
دوره نگار ۱۴۶
روش ولش ۱۴۶

فصل پنجم: فیلترینگ دیجیتال

مشخصات فیلتر ۱۴۹
فیلتر پایین گذر ۱۵۰
فیلتر بالا گذر ۱۵۲
فیلترهای میان گذر و میان ناگذر ۱۵۳
فیلترینگ مستقیم در حوزه فرکانس ۱۵۵
زمان فیلترینگ دامنه ۱۵۷
فیلترهای FIR ساده ۱۵۸
افزایش مرتبهٔ فیلتر ساده ۱۵۹
طراحی BPF با استفاده از فیلتر جمع و اختلاف ۱۶۱
طراحی فیلتر FIR با استفاده از روش پنجره ۱۶۲
طراحی فیلتر IIR ۱۶۸

فصل ششم: استخراج ویژگی‌ها

استخراج ویژگی‌ها ۱۷۲
ویژگی‌های ساده ۱۷۲
همبستگی ۱۷۴
مدل خودرگرسیو ۱۷۵
انتخاب مرتبه مدل اتورگرسیو ۱۷۶
مدل خودرگرسیو برای پیش‌بینی مقادیر سیگنال ۱۷۷
ضرایب خود رگرسیو به عنوان ویژگی‌های قابل تبعیض وظایف ذهنی ۱۷۸
ویژگی‌های طیفی – تابع چگالی توان ۱۷۹
ویژگی‌های استخراج شدهٔ تابع چگالی توان ۱۸۰
نسبت عدم تقارن PSD ۱۸۰
همبستگی یا انسجام طیفی ۱۸۲
پیک‌های طیف ۱۸۲
محاسبه تابع چگالی توان با استفاده از ویژگی‌های AR ۱۸۳
توصیف‌های جورت ۱۸۵
ویژگی‌های دامنه زمان ۱۸۵
ویژگی‌های زمان– فرکانس مشترک ۱۸۶

فصل هفتم: روش‌شناسی طبقه‌بندی

طبقه‌بندی چیست؟ ۱۸۹
طبقه‌بندی نزدیکترین همسایه ۱۹۰
الگوریتم K-NN ۱۹۱
مزایا و معایب طبقه‌بندی K نزدیکترین همسایه ۱۹۳
برنامه متلب K-NN ۱۹۴
کاهش اندازهٔ مجموعه دادهٔ آموزش K-NN ۱۹۵
نزدیکترین همسایه فشرده ۱۹۶
نزدیکترین همسایه ویرایش شده ۱۹۸
نرون مصنوعی ۱۹۹
شبکه عصبی پرسپترون چند لایه ۲۰۰
معماری شبکه عصبی پرسپترون چند لایه پس انتشار ۲۰۱

آموزش پرسپترون چند لایه پس انتشار ۲۰۱
آزمون کارایی طبقه‌بند پرسپترون چند لایه پس انتشار۲۰۳
پیاده‌سازی پرسپترون چند لایه پس انتشار با متلب ۲۰۳
مشکلات پرسپترون چند لایه پس انتشار ۲۰۴
معیارهای کارایی ۲۰۶
وارسی اعتبار ۲۰۷
وزن کلاس برابر.....۲۰۷
حذف یک خروجی ۲۰۸
معیارهای آماری برای مقایسهٔ دو مدل ۲۰۸
آزمون فرضیه‌ها.....۲۰۹

فصل هشتم: فصل ارائهٔ مثال‌های فیزیولوژیکی
خواندن داده‌ها از فایل‌ها.....۲۱۲
خواندن داده‌ها از فایل‌های موجود در پایگاه داده‌ها ۲۱۲
خواندن داده‌ها از فایل‌های موجود در سیستم ۲۱۳
پیدا کردن پیک‌ها در سیگنال‌های اشباع یا کوتاه شده ۲۱۳
حذف نویز از سیگنال ۲۱۵
استفاده از تحلیل مؤلفه‌های مستقل برای حذف آرتیفکت‌ها از ECG ۲۱۹
تحلیل پیک‌ها در سیگنال ECG ۲۲۱
اندازه‌گیری پیک‌های دامنه در سیگنال ECG ۲۲۱
شبیه‌سازی سیگنال ECG.....۲۳۱
ویژگی‌های قابل توجهی از شکل موج ECG ۲۳۱
سری فوریه ۲۳۲
پیاده‌سازی در متلب ۲۳۳
منابع و مآخذ ۲۵۰

پیشگفتار

تجزیه و تحلیل سیگنال‌های بیولوژیکی شامل چندین موضوع میان رشته‌ای است که با تحلیل سیگنال‌های تولید شده توسط فرایندهای بیولوژیکی مختلف بدن انسان مرتبط هستند. معمولاً آنالیز این سیگنال‌ها در توضیح یا شناسایی شرایط پاتولوژیک بدن انسان بسیار مفید است. سیگنال‌های بیولوژیکی در اولیه‌ترین شکل خود، اطلاعات زیادی نمی‌دهند، لذا جهت استخراج اطلاعات مربوط به این سیگنال‌ها باید تحلیل‌های لازم صورت بگیرد. این تحلیل‌ها با تلاش سازمان‌های مراقبت بهداشتی مدرن به منظور تأمین هزینه مؤثر تشخیص و درمان دارای اهمیت شده است. قدرت محاسباتی سریعی که در سال‌های اخیر ایجاد شده است، روش‌هایی را برای تحلیل فرایندهای پیچیده فراهم آورده است.

هدف اصلی و اولیهٔ از این کتاب فراهم آوردن دانش و شناخت کافی از تکنیک‌های پردازش سیگنال و الگوریتم‌های دسته‌بندی برای تحلیل سیگنال‌های بیولوژیکی برای افراد علاقه‌مند به این حوزه‌ها است. مطالب ارائه شده در این کتاب دانشی از اصول اولیه و اساسی مربوط به پردازش سیگنال‌های دیجیتال را برای خوانندگان فراهم می‌آورد.

این کتاب، مطالبی از قبیل آموزش نرم‌افزار متلب، آشنایی با فیزیولوژی و ویژگی‌های سیگنال‌های بیولوژیکی مختلف، توصیف و به کارگیری تکنیک‌های مختلف پیش و پسا پردازش از قبیل: آماده‌سازی، فیلترینگ، استخراج ویژگی‌ها، دسته‌بندی و روش‌های اعتبارسنجی آماری برای سیگنال‌های بیولوژیکی، حل مشکلات عملی مربوط به سیگنال‌های بیولوژیکی با متلب و ارائهٔ مثال‌های فیزیولوژیکی با متلب را پوشش داده است.

کتاب حاضر به عنوان کتابی جامع برای آموزش پردازش سیگنال‌های بیولوژیکی به افراد علاقه‌مند در حوزه‌های مختلف آکادمیک، صنعتی و تدریس در دانشگاه توصیه می‌شود و برای افرادی که هیچ گونه دانش زمینه‌ای از پردازش سیگنال ندارند بسیار کارا و پربار خواهد بود؛ زیرا این افراد می‌توانند با صرف کمترین زمان بیشترین بهره را کسب نمایند.

در این اثر سعی شده که جامعیت و اختصار در کنار هم رعایت شود. نظم مطالب بر طبق مطالعات و دانش نویسندگان تنظیم و سعی شده است تا مطالب به نحوی فصل‌بندی و سازماندهی یابد که نظم منطقی مطالب به آشفتگی و سردرگمی بدل نشود. در نگارش مطالب تلاش شده است تا سخن رسول اکرم (ص) را که فرمودند: «رَحِمَ اللهُ اِمرَأً عَمِلَ عَمَلاً فَاَحکَمَه» چون چراغی، روشنی‌بخش راهمان قرار دهیم تا دستیابی به هدفی که مد نظر بود، میسر گردد تا اثری سودمند و پربار بر جای گذاشته شود. بی‌شک این کتاب همانند هر مکتوب بشری عاری از نقص و کاستی نخواهد بود، امیدواریم که صاحب‌نظران فرهیخته را از ریزبینی خود بی‌بهره نگذارند و آنچه را که در خور بیان دیدند به آدرس moulaei.kh@tak.iums.ac.ir ارسال نمایند.

در پایان نیز بر خود لازم می‌دانم از پدر و مادر بزرگوارم و تمامی کسانی که به نوعی مشوق من در امر تألیف این کتاب بوده‌اند، تشکر و قدردانی نمایم.

خدیجه مولایی

تابستان ۱۳۹۶

فصل اول
آموزش نرم‌افزار متلب

⊡ معرفی نرم‌افزار متلب

نرم‌افزار متلب برنامه کامپیوتری است برای کسانی که با محاسبات عددی، بویـــژه جـبر خطی سر و کار دارد. نام این نرم‌افزار از عبارت انگلیسی LABoratory MATrix اقتباس شده و هدف اولیه آن قادر ساختن مهندسین و دانشمندان به حـــل مسـائل مختلف شـامل عملیـات ماتریسی بدون نیاز به نوشتن برنامه در زبان‌های برنامه‌نویســـی متـداول همچـون C و Fortran است. با گذشت زمان قابلیت‌های بسیار بیشتری به این نرم‌افزار افزوده شده است به طـــوری کـه در حـال حاضر متلب به ابزار پر قدرتی برای ترسیم داده‌ها، برنامه‌نویسی و انجام محاسبات مهندسـی و پژوهشی تبدیل شده است.

متلب محیط نرم‌افزاری برای انجام محاسبات عددی و یک زبان برنامه‌نویسی نسل چهارم است. واژه متلب هم به معنی محیط محاسبات رقمی و هم به معنی خود زبان برنامه‌نویسی مربوطه است که از ترکیب دو واژ ماتریس[1] و آزمایشگاه[2] ایجاد شده‌است. این نام حاکی از رویکرد ماتریس محور برنامه است که در آن حتی اعداد منفرد هم به عنوان ماتریس در نظر گرفته می‌شوند.

ساخت رابط گرافیکی کاربر مانند دیالوگ‌هایی که در محیط‌های ویژوال مانند بیسیک و C وجود دارند، در متلب نیز امکان‌پذیر است. این قابلیت، ارتباط بهتری را میان برنامه‌های کاربردی نوشته شده با متلب و کاربران برقرار می‌کند. متلب که از محصولات شرکت MathWorks است، برای گروه‌های مختلف مهندسی در رشته‌های مختلف از جمله مهندسی پزشکی، برق، مکانیک، رایانه و غیره کاربرد بسیاری دارد.

به طور پیش فرض در رابط متلب ۴ قسمت وجود دارد: Command window در سمت راست، command history در سمت چپ پایین، workspace در سمت چپ بالا که با تب از هم جدا شده‌اند و current directory در سمت چپ پایین. همچنین در سمت چپ پایین دکمه start قرار دارد که مانند محیط ویندوز از طریق آن می‌توانید به shortcutهای متلب دسترسی داشته باشید. Command

1. Matrix
2. Laboratory

window جایی است که در آن دستورات تایپ می‌شوند تا متلب بتواند نتایج را محاسبه، ارزیابی و یا به صورت نمودار رسم کند. دستوراتی که در command window اجرا می‌کنیم، در command history بایگانی می‌شوند. از مزایای این بخش می‌توان به دو مورد زیر اشاره کرد:

۱- می‌توان یک دید کلی از دستوراتی که قبلا اجرا شده‌اند به دست آورد.

۲- زمان اجرای دستورات ذخیره می‌شود.

می‌توانید روی هر دستور کلیک راست کرده، گزینه کپی را انتخاب و در command window قرار داده و اجرا کنید. یا با دوبار کلیک روی دستور، آن را اجرا کنید. به طور نمونه برای اجرای یک دستور به ماتریسی نیاز دارید که از طریق محاسبات قبلی به دست آمده و اکنون از حافظه پاک شده است. می‌توانید آن دستور را سریعاً از طریق command history اجرا کنید تا داده مورد نیاز در workspace ذخیره شود. البته نباید فراموش کرد که هنگام اتمام کار متغیرها باید ذخیره شوند. می‌توان با دو بار کلیک کردن بر روی متغیرهای تعریف شده، تک‌تک مؤلفه‌های آنها را ویرایش کرد. همچنین اگر بر روی متغیرها کلیک راست شود، می‌توان هر متغیر را به شکل‌های مختلف رسم کرد.

هر فایل جدیدی که در متلب ایجاد می‌شود در current directory ذخیره می‌شود. ممکن است بخواهید آدرس curret directoy را تغییر دهید یا فایلی را از آدرسی دیگر اجرا کنید برای ایجاد یک current directry جدید چند روش وجود دارد. می‌توان از طریق پنجره current directory مانند پنجره‌های محیط ویندوز عمل کرد و در بخش toolbar در قسمت curret directory روی دکمه "..." کلیک کنید و آدرس را قرار دهید. همچنین می‌توانیم دستور مربوطه را در command window تایپ کنیم. در Current directory می‌توان فایل‌ها را اجرا کرد و تغییر داد. ولی اگر بخواهید یک فایل فقط اجرا شود باید آدرس آن را add کنید. برای add کردن یک directory از دستور زیر استفاده می‌شود. به عنوان مثال در درایو D پوشه‌ای با نام matlab ایجاد شده است که آدرس آن به صورت زیر add می‌شود.

کد
```
>> addpath D:\matlab
```

بهترین روش برای یادگیری متلب این است که از دستورات ساده شروع کنید و نتایج آن را مشاهده کنید، هرگاه دستوری را در متلب اشتباه وارد کنید، متلب پیغام خطایی در پنجره Command نمایش می‌دهد که نوع خطا و همچنین محل خطا را برای شما مشخص می‌کند. پس با دستورات ساده شروع کنید، نتایج و یا پیام‌های خطا را مشاهده کرده و بتدریج به سراغ دستورات پیچیده‌تر بروید. برای نوشتن دستورات در پنجره command، ابتدا در پنجره Command کلیک کنید تا فعال شود، سپس می‌توانید دستورات مورد نظر خود را وارد کنید و با فشار دادن کلید enter از صفحه کلید، نتیجه اجرای دستورات را در همان پنجره مشاهده کنید.

مثال: دستور ساده زیر را وارد کنید:

```
4+7
```

نتیجه به صورت زیر در پنجره command نمایش داده می‌شود:

```
ans=
11
```

ans نشان‌دهندهٔ ابتدای کلمه answer می‌باشد. هنگامی که در متلب برای نتیجه یک محاسبه نامی انتخاب نشده باشد، نرم‌افزار متلب نام ans را برای آن انتخاب می‌کند، یعنی این که متغیری به نام ans با مقدار ۲ ایجاد می‌شود. چنانچه برای محاسبات بعدی به این عدد احتیاج داشته باشید، باید حتما نام دیگری برای آن انتخاب کنید مثلاً (b= 4+7) زیرا نرم‌افزار متلب به دستور بعدی که نام متغیری برای آن در نظر گرفته نشده باشد دوباره نام ans قبلی را اختصاص می‌دهد و عملاً مقدار آن پاک می‌شود.

نکته: چنانچه تمایل داشته باشید که دستورات قبلی و نتایج آنها را که در پنجره Command نمایش داده شده‌اند پاک کنید، تنها کافی است که بر روی قسمتی از پنجره Command کلیک سمت راست کرده و گزینه Clear Command Window را انتخاب نمایید. باید دقت داشته باشید که با این کار تنها دستورات و نتایج نشان داده شده در پنجره Command پاک می‌شوند اما، متغیرهایی که توسط این دستورات در متلب تعریف شده‌اند، همچنان وجود دارند و می‌توان از آنها استفاده نمود. دستوری که متغیرها را به طور کامل از متلب پاک می‌کند، دستور clear می‌باشد. از طرفی چنانچه برنامه متلب را ببندید، تمامی متغیرهای تعریف شده در متلب پاک می‌شوند و دفعه بعد که متلب را باز کنید هیچ متغیری در آن تعریف نشده است.

اگر بخواهید نتیجه اجرای خطی از دستورات در پنجره Command نمایش داده نشود تنها کافی است که در پایان آن خط از دستورات، علامت ((;)) را بنویسید. با به کار بردن این علامت در پایان هر خط، متلب دستورات آن خط را اجرا کرده و نتیجه محاسبات را در متغیرها ذخیره می‌کند، اما نتیجه محاسبات را در پنجره Command نمایش نخواهد داد.

⊡ M-file

مجموعه‌ای از دستورات متلب را می‌توانید در یک پرونده ذخیره کنید و سپس آنها را یک‌جـا اجرا نمایید. اگر بخواهید چنین پرونده‌ای در محیط متلب قابل اجـرا باشـد بـاید حتمـاً دارای دنباله "m". باشد. در صورتی که از ویرایشگر[1] متلب اسـتفاده کنیـد، دنباله "m ". به طور خودکار در

1. MATLAB Editor

هنگام ذخیره پرونده به نام آن افزوده می‌گردد. در صورت اســتفاده از ویرایشگر دیگری غیر از ویرایشگر متلب (نظیر Notepad) اطمینان حاصل کنید کــه پرونـده حتما به روش ascii و با دنباله "m ". ذخیره شود. به عبارتی دیگر، برای مسائل پیچیده ابزارهای command window و command history ناکافی هستند. راه حل بهتر آن است که یک M-file ایجاد شود. دو نوع M-file وجود دارد. Script M-file و function M-file.

M-fileها، فایل‌های متنی معمولی حاوی دستورات متلب هستند. می‌توان با استفاده از نرم‌افزارهای کار با متن مانند word و wordpad و notepad و ... M-fileها را ایجاد و یا ویرایش کرد. همین طور می‌توان از editor خود متلب برای این کار استفاده کرد که امکانات بیشتری در اختیار قرار خواهد داد. برای ایجاد یک M-file می‌توان در command window واژهٔ edit را تایپ کرد یا اینکه از طریق file < new <m-file آن را ایجاد کنید.

❐ Script M-file

برای ایجاد یک script M-file، ابتدا دستور زیر را در M-file تایپ کنید و آن را با نام m1task. در current directory خود ذخیره کنید.

کد

```
format long
x = [0.1, 0.01, 0.001];
y = sin(x)./x
```

برای اجرای این فایل باید در command window، task1. را تایپ کنید. اگر M-file را با استفاده از نرم‌افزاری غیر از متلب ایجاد شود باید یادآوری نمود که فرمت فایل باید m. قرار داده شود. اگر M-file را در پنجره‌ای غیر از command window ذخیره کردید و در هنگام اجرا هشدار دریافت شد باید به توضیحات مربوط به current directory در صفحات قبل مراجعه شود. در نسخه‌های جدید متلب از همان پنجره ادیتور می‌توانید دستورات را اجرا کنید. در پنجره toolbars یک دکمه سبز رنگ مثلثی وجود دارد که در صورت کلیک کردن بر روی آن ابتدا برای save کردن، آدرس می‌خواهد و سپس آن را اجرا می‌کند. در نتیجه خروجی در command window چاپ می‌شود. حالا می‌توانید M-file خود را ویرایش کنید. برای مثال اگر بخواهید y را به ازای x = 0.0001 به دست آورید تغییرات زیر نیاز است.

کد:

```
format long
x = [0.1, 0.01, 0.001, 0.0001];
y = sin(x)./x
```

❐ Echo کردن دستورات

همان طور که دیده می‌شود، دستورات M-file در command window چاپ نمی‌شود. برای چاپ هم‌زمان دستورات و خروجی از echo استفاده کنید.

کد:
```
echo on
format long
x = [0.1, 0.01, 0.001];
y = sin(x)./x
echo off
```

❐ ساختار script M-file

برای آنکه نتایج script M-fileها مجدداً قابلیت تولید داشته باشند، نتایج باید در همه جا یکسان باشد، متغیرهای تعریف شده قبلی روی نتیجه تأثیر نگذارد و پنجره‌های گرافیکی قبلی در نتیجه فعلی تأثیر نگذارد. همچنین باید نکات زیر مد نظر قرار بگیرند.

در ابتدای script، clear all را تایپ کنید تا مطمئن شوید که متغیرهای تعریف شده قبلی روی نتایج بی‌تأثیرند. همچنین می‌توانیم دستور close all را در ابتدای script M-fileهایی قرار دهیم که نمودار تولید می‌کنند. با این دستور تمام پنجره‌های گرافیکی که قبلاً باز شده‌اند، بسته می‌شود. مثال زیر را با به کارگیری مفاهیم بالا آشنا خواهد کرد.

کد:
```
% Remove old variable definitions
clear all
% Remove old graphics windows
close all
% Display the command lines in the command window
echo on
% Turn on 15 digit display
format long
% Define the vector of values of the independent
variable
x = [0.1, 0.01, 0.001];
% Compute the desired values
y = sin(x)./x
% These values illustrate the fact that the limit of
% sin(x)/x as x approaches 0 is equal to 1.
echo off
```

گاهی اوقات چه در command window و چه در M-file دستور به حدی طولانی است که در یک سطر جا نمی‌شود. در این موارد می‌توانیم در انتهای خط با یک فاصله از دستورات، خط را تایپ

و در ادامه کلید Enter را فشار دهیم. حال می‌توانید ادامه خط را در خط پایین بنویسید. با این کار در خط بعدی علامت << نمی‌آید.

❐ تابع M-file

گاهی اوقات لازم است که فرایندی برای تعداد زیادی ورودی به یک پارامتر یا متغیر تکرار شود. در این صورت می‌توانیم به یک تابع پیش فرض متلب چندین ورودی اعمال کنیم و خروجی بگیریم. در متلب با استفاده از دستور inline می‌توان تابع ساخت، اما ساخت تابع با استفاده از m-file انتخابی حرفه‌ای‌تر می‌باشد. معمولاً برنامه‌نویسان حرفه‌ای، چارچوب برنامه را در یک فایل اصلی می‌نویسند و بخش‌های مختلف برنامه را به صورت توابعی می‌نویسند که یک یا چند ورودی را دریافت کرده و محاسبات لازم را انجام می‌دهند و سپس یک یا چند خروجی را بر می‌گردانند. برنامه‌نویس، عملکرد هر تابع را جداگانه کنترل می‌کند و سپس زمانی که اطمینان حاصل شد، تمرکز اصلی را بر روی فایل اصلی که حاوی چارچوب برنامه می‌باشد، گذاشته می‌شود و تنها در فایل اصلی ارجاعاتی به توابع ساخته شده ارائه خواهد شد.

یکی از ساختارهای مهم در تمام زبان‌های برنامه‌نویسی، توابع هستند. علاوه بر توابعی که به صورت آماده در متلب وجود دارد، می‌توان توابعی را به منظور انجام محاسبات مورد نظر ایجاد کرد. یک تابع یک یا چند داده را در ورودی دریافت می‌کند و پس از انجام محاسبات لازم نتایج را در قالب یک یا چند متغیر خروجی به شما بر می‌گرداند. خط اول یک تابع که خط تعریف تابع نیز نامیده می‌شود باید از ترتیب زیر پیروی نماید.

نکته: با استفاده از توابع می‌توان چندین دستورالعمل یا یک قطعه برنامه را نامگذاری کرد و در مواقع مورد نیاز اجرا نمود.

در متلب توابع به صورت زیر نوشته می‌شود:

```
function name(آرگومان)
دستورات
End
```

name که نام تابع و دلخواه است از قاعده نامگذاری شناسه‌ها تبعیت می‌کند و بایستی تابع را به صورت یک file-M نوشت و با نامی دقیقاً مشابه name ذخیره نمود. ممکن است در یک تابع آرگومان مورد نیاز باشد یا می‌توان در صورت عدم نیاز از نوشتن آن صرف نظر کرد.

با مثالی ساده، نحوه ساخت یک تابع با استفاده از m-file را شرح می‌دهیم:

مثال: می‌خواهیم تابعی به نام add بسازیم که در ورودی، دو عدد را دریافت کرده و سپس مجموع آن دو عدد را در خروجی برگرداند. ابتدا یک m-file خالی باز می‌کنیم و کد زیر را در آن می‌نویسیم:

```
function [z] = add(x,y)
z=x+y;
```

در کد بالا، کلمه function اعلام می‌کند که این m-file به صورت یک تابع می‌باشد. خروجی با نام z و دو ورودی با نام‌های x و y مشخص شده‌اند. کلمه‌ای که بعد از علامت تساوی قرار می‌گیرد (در اینجا کلمه add)، نام تابع را مشخص می‌کند و باید در مرحله بعد، m-file را دقیقا با همین نام ذخیره کنیم. بنابراین m-file را با نام add ذخیره می‌کنیم.

همان طور که مشاهده کردید، خط اول، ساختار تابع را مشخص می‌کند و در خط‌های بعدی، تنها کاری که باید انجام دهیم، این است که روابط بین ورودی‌ها و خروجی را تعریف کنیم. مثال بالا تنها یک جمع ساده می‌باشد.

پس از ذخیره کردن m-file، دیگر کاری با آن نداریم. باید دقت کنید که اگر بخواهید این m-file را با گزینه Save and run در بالای پنجره اجرا کنید، با یک خطا مواجه می‌شوید، زیرا این m-file یک تابع است و باید ورودی‌هایی برای آن تعریف کنیم. اکنون باید تابع ساخته شده را آزمایش کنیم. دستور زیر را در پنجره Command می‌نویسیم (و یا در یک m-file دیگر):

```
B=2;
C=3;
A=add(B,C)
```

نتیجه:

```
A =
    5
```

نکته: باید m-file تابع درون Current folder و یا سایر مسیرهای مشخص شده برای نرم‌افزار متلب باشد. در غیر این صورت، متلب یک پیام خطا را نمایش می‌دهد. این پیام خطا به این دلیل است که متلب نمی‌تواند m-file با نام add را پیدا کند.

نکته: در دستور فوق مشاهده کردیم که نام‌هایی که برای ورودی و خروجی تابع add استفاده کردیم (A، B و C)، با نام‌های درون خود کدهای تابع add، متفاوت است (x، y و z). بنابراین نیازی به نگرانی درباره نام‌های به کار رفته در ساختار تابع نیست و از هر نام دلخواهی برای ارجاع به تابع می‌توانید استفاده کنید.

⊡ اضافه کردن توضیحات[1]

در متلب می‌توانیم توضیحاتی را برای هر خط دستور قرار دهیم. این توضیحات می‌توانند محاسبات را شرح دهند و حتی نتیجه را تفسیر کنند. هر خطی که در script M-file با علامت " ٪ " شروع شود، یک توضیح است و متلب با آن به عنوان دستور رفتار نمی‌کند. در کد زیر فایل task 1m . را با اضافه کردن توضیح ویرایش کرده‌ایم.

کد:

```
echo on
% Turn on 15 digit display
format long
x = [0.1, 0.01, 0.001];
y = sin(x)./x
% These values illustrate the fact that the limit of
% sin(x)/x as x approaches 0 is 1.
echo off
```

زمانی که می‌خواهید به script M-file یک توضیح اضافه کنید به یاد داشته باشید که در ابتدای سطر علامت ٪ را قرار دهید. اگر دستورات را echo کنید در command window توضیحات نیز ظاهر می‌شوند.

⊡ استفاده از دستور demo

دستور demo پنجره جدیدی باز می‌کند که شما در آن می‌توانید مثال‌های متعددی از امکانات متلب را بیابید. بسیاری از این مثال‌ها، نمودارهای جالب و همراه با جزئیات تولید می‌نمایند و همچنین توضیحات مفیدی را درباره نحوه استفاده از متلب ارائه می‌دهند. توصیه می‌شود که حتماً تعدادی از این مثال‌ها را مشاهده کنید تا متوجه شوید که چه کارهایی می‌توان با متلب انجام داد. بویژه دقت کنید که چگونه برنامه‌های ساده می‌توانند نتایج پیچیده‌ای تولید نمایند.

⊡ پنجره Help در متلب

یکی از مهمترین امکانات نرم‌افزار متلب استفاده از Help می‌باشد. شما می‌توانید هر تابع دلخواه خود را در Help جستجو کرده و مثال‌هایی از آن را مشاهده نمایید.

گاهی اوقات، کارکردن با Help متلب می‌تواند بسیار دشوار باشد و آن زمانی است که شما می‌خواهید دستوری را برای یک عمل خاص بیابید، اما نام آن دستور را نمی‌دانید و مجبورید با

1. Comments

کلماتی که آن عمل خاص را توصیف می‌کنند به جستجو بپردازید. به دلیل حجم زیاد دستورها و ابزارهای مختلف نرم‌افزار متلب، باید بتوانید کلمه درست را بیابید و با آن کلمه نام دستور مورد نظر را پیدا کنید. اما زمانی که نام دستور را بدانید، Help متلب، مرجع بسیار مناسبی است، زیرا نحوه دقیق استفاده از آن دستور و همچنین دستورهای با عملکرد مشابه آن دستور را به شما نشان خواهد داد (دستورهای مرتبط با آن).

❐ باز کردن پنجره Help

در پنجره اصلی نرم‌افزار متلب از منوی Help، گزینه Product Help را انتخاب کنید. پنجره Help متلب باز می‌شود. در پنجره Help، قسمتی برای جستجو وجود دارد که می‌توانید با آن کلمه یا کلمات مورد نظر خود را وارد کرده و به جستجو در صفحات Help متلب بپردازید.

❐ نحوه دیدن صفحات Help با دستور Help

تنها کافی است که در پنجره Command تایپ کنید:

```
Help
```

و سپس کلید enter از صفحه کلید را فشار بدهید. لیستی از صفحات Help متلب به صورت لینک در پنجره Command نمایش داده می‌شود و می‌توانید بر روی آنها کلیک کرده و مطالب مورد نظر خود را مشاهده کنید. اگر بخواهید Help مربوط به یک دستور خاص را ببینید، باید پس از کلمه Help، نام آن دستور را بنویسید. به مثال زیر توجه کنید:

اگر فرض کنید بخواهید Help مربوط به دستور elfun را مشاهده کنید، برای این منظور می‌نویسید:

help elfu نتیجه این دستور باعث می‌شود که لیستی از توابع مقدماتی ریاضی نمایش داده شود. لیست نمایش داده شده بسیار بلند است و به بخش‌های مثلثات و اعداد مختلط و توابع گرد کردن و باقی‌مانده‌ها، تقسیم شده است.

برای اینکه بدانیم که یک تابع خاص چه کاری را انجام می‌دهد و همچنین نحوهٔ فراخوانی آن به چه شکل می‌شود، کافی است ابتدا کلمهٔ help و سپس نام آن تابع را در متلب تایپ شود (و کلید Enter را فشار دهیم). به عنوان مثال دستور زیر باعث می‌شود تا توضیحات تابع (sin) نمایش داده شوند:

```
help sin
```

برخی از اطلاعات بالا در پنجره Command از نرم‌افزار متلب به صورت لینک نمایش داده می‌شوند و چنانچه بر روی آنها کلیک شود، اطلاعات بیشتری نمایش داده خواهد شد.