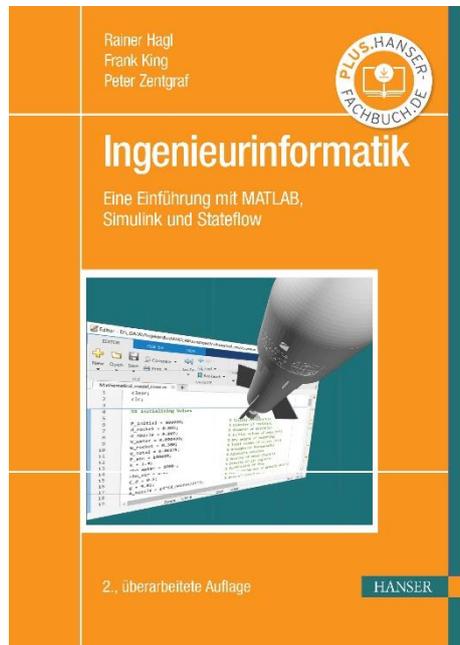


HANSER



Leseprobe

zu

Ingenieurinformatik

von Rainer Hagl, Frank A. King und Peter Zentgraf

Print-ISBN: 978-3-446-47515-1
E-Book-ISBN: 978-3-446-47730-8

Weitere Informationen und Bestellungen unter
<https://www.hanser-kundencenter.de/fachbuch/artikel/9783446475151>

sowie im Buchhandel

© Carl Hanser Verlag, München

Vorwort

Dieses Lehrbuch stellt eine Einführung in grundlegende Themen der Ingenieurinformatik dar. Schwerpunkt ist die zeiteffiziente Analyse und der Entwurf von technischen Systemen im Ingenieurbereich mittels Software aus dem Bereich der computerunterstützten Entwicklung (Computer Aided Engineering, CAE). Das Thema Ingenieurinformatik wird beispielhaft anhand der Entwicklungsumgebung MATLAB, Simulink und Stateflow zur Analyse technischer Systeme und zur Entwicklung elektronischer Steuergeräte verdeutlicht. Diese Entwicklungsumgebung ist in der Industrie weltweit quasi als Standard für die beschriebenen Aufgaben etabliert. Dadurch ist ein hoher Praxisbezug gegeben.

Das Fachbuch ist insbesondere für die Bachelorausbildung von Studierenden der Ingenieurwissenschaften in folgenden Studienschwerpunkten konzipiert:

- Elektro- und Informationstechnik
- Mechatronik
- Maschinenbau
- Automatisierungstechnik
- Energietechnik
- Gebäudetechnik

Es eignet sich ebenso für technisch Interessierte, die sich in die Thematik einarbeiten wollen.

Zunächst wird in der Einführung auf die Entwicklungen eingegangen, die zum heutigen Einsatz von Computern geführt haben. Es wird die weite Verbreitung der computerunterstützten Entwicklung und Fertigung im Ingenieurbereich dargestellt. Grundlegende Kenntnisse, Zusammenhänge und Werkzeuge bei der Softwareentwicklung werden auf Basis der im technischen und naturwissenschaftlichen Bereich weit verbreiteten Sprache MATLAB gezeigt. Diese sind allgemein gültig und lassen sich weitestgehend auf andere Programmiersprachen übertragen. In vielen Projekten ist es hilfreich, grafische Bedienoberflächen zu verwenden. Dieses Themenfeld wird am Beispiel der in MATLAB eingebauten Möglichkeiten dargestellt. Die digitale Arbeitsweise von Computern führt zu mehr oder weniger starken Begrenzungen des Wertebereiches von Zahlen und damit einhergehend auch zu Einschränkungen der Berechnungsergebnisse. Insbesondere bei der Entwicklung elektronischer Steuergeräte, bei denen die Herstellkosten ein wichtiger Faktor sind, ist die Wertebereichsbegrenzung ein wichtiger Punkt bei der Entwicklung. Eine Einführung hierzu wird im Kapitel „Zahlenformate“ gegeben. Viele Berechnungsaufgaben können aufgrund ihrer Komplexität nur auf Computern mit Näherungsverfahren gelöst werden. Grundlegende Zusammenhänge werden im Kapitel „Numerische Integration“ behandelt.

Eine wichtige praktische Anwendung ist die Simulation dynamischer Systeme, welche beispielhaft anhand des Softwarepaketes Simulink gezeigt wird. In vielen Steuerungsaufgaben muss auf Ereignisse mit vorbestimmten Aktionen reagiert werden. Ereignisdiskrete Systeme werden daher in einem separaten Kapitel behandelt. Um die Arbeitsweise praxisnah dazustellen, wird in diesem Kapitel das Softwarepaket Stateflow verwendet. Manche Berechnungsaufgaben müssen aufgrund der vielen Einzelschritte auf mehrere Recheneinheiten verteilt werden, um in akzeptablen Wartezeiten Ergebnisse zu erhalten. Daher wird in einem eigenen Kapitel auf die Thematik „Paralleles Rechnen“ in Grundzügen eingegangen. Mathematische Umformungen und Vereinfachungen können computerunterstützt erfolgen. Im Kapitel „Symbolisches Rechnen“ wird einführend auf diese Möglichkeit eingegangen. Den Kapiteln zugeordnete Übungen erlauben eine Überprüfung des Lernfortschrittes. Weiteres Zusatzmaterial steht den Lesern unter *plus.hanserfachbuch.de* zur Verfügung. Den Zugangscode finden Sie auf der ersten Seite des Buches.

In den Jahren seit dem Erscheinen der ersten Auflage blieb die Zeit nicht stehen. Augenfälligste Veränderung ist der nun geschlechtergerechte Buchtitel sowie das Hinzukommen zweier Autoren. Auch konnten von den Autoren weitere Erfahrungen beim Einsatz des Buches in Vorlesungen und Übungen gesammelt werden. Diese sind, zusammen mit technischen Weiterentwicklungen, in die vorliegende zweite Auflage eingeflossen. Dies betrifft insbesondere das Kapitel zu grafischen Bedienoberflächen (Apps), dessen Hauptteil auf aktuellem Stand neu aufgebaut wurde.

Eine Vielzahl konstruktiver Rezensionen der ersten Auflage hat zur Beseitigung von Fehlern und inhaltlichen Klarstellungen in der zweiten Auflage geführt. Hierfür möchten sich die Autoren und der Verlag herzlich bedanken. Allerdings konnten nicht alle Anregungen berücksichtigt werden, da diese zum Teil konträr waren. Bitte haben Sie dafür Verständnis.

Auch in dieser zweiten Auflage haben sich sicherlich noch Fehler eingeschlichen. Vielleicht ist das eine oder andere auch nicht ganz verständlich. Über Rückmeldungen zu Fehlern oder Verbesserungsvorschläge würden wir uns sehr freuen, da diese zu einer kontinuierlichen Verbesserung führen. Sie können uns diesbezüglich gerne eine E-Mail an

frank.king@th-rosenheim.de

peter.zentgraf@th-rosenheim.de

senden. Für Ihre Unterstützung möchten wir uns bereits im Voraus bei Ihnen bedanken.

März 2023

Rainer Hagl | Frank A. King | Peter Zentgraf

■ Danksagung

Für die zahlreichen konstruktiven Diskussionen und Anregungen rund um die Ausbildung im Bereich der Ingenieurinformatik und Simulation möchten wir uns bei unseren Kollegen Prof. Franz Perschl und Prof. Martin Versen bedanken. Daneben gebührt noch immer ein herzlicher Dank allen, die durch ihr Zutun am Gelingen der vorhergehenden ersten Auflage beteiligt waren.



Der Verlag und die Autoren haben sich mit der Problematik einer gendergerechten Sprache intensiv beschäftigt. Um eine optimale Lesbarkeit und Verständlichkeit sicherzustellen, wird in diesem Werk auf Gendersternchen und sonstige Varianten verzichtet; diese Entscheidung basiert auf der Empfehlung des Rates für deutsche Rechtschreibung. Grundsätzlich respektieren der Verlag und die Autoren alle Menschen unabhängig von ihrem Geschlecht, ihrer Sexualität, ihrer Hautfarbe, ihrer Herkunft und ihrer nationalen Zugehörigkeit.

Zum Cover des Buches:

Der abgebildete Programmcode und die animierte Rakete wurden in dem Projekt „water rocket“ an der Technischen Hochschule Rosenheim im Jahr 2021 unter der Leitung von Prof. Zentgraf entwickelt. Die Projekt-Unterlagen und der zugehörige fünfminütige Filmbeitrag können unter

<https://www.th-rosenheim.de/die-hochschule/labore/regelungstechnik/water-rocket>

eingesehen werden.

Inhalt

■	Hinweis	13
■	Formelsymbole	14
■	Programmbeispiele	15
1	Einführung	17
	1.1 Historie Rechenmaschinen	20
	1.2 Computerunterstützung bei der Lösung mathematischer Aufgaben	27
	1.3 Modellbasierte Steuergeräteentwicklung	31
2	Grundlagen der Programmierung	37
	2.1 Erste Schritte in MATLAB und Grundregeln	38
	2.1.1 Bedienoberfläche	38
	2.1.2 Wertezuweisung und Variablendefinition	41
	2.1.3 Hilfeunterstützung und elektronische Dokumentation	46
	2.1.4 Ein- und mehrdimensionale Felder	49
	2.1.5 Arithmetische Operatoren für den Einstieg	51
	2.1.6 Relationale und logische Operatoren	53
	2.1.7 Sonderzeichen	55
	2.1.8 MATLAB Editor	57
	2.1.9 Programmbeispiel	62
	2.1.10 Script und Function	66
	2.1.11 Workspace und Gültigkeitsbereich von Variablen	75
	2.1.12 Arbeitsverzeichnisse	77
	2.1.13 Fehlersuche und Debugger	80
	2.1.14 Freigabe und Initialisierung von Speicherbereichen	84
	2.1.15 MATLAB Version	85
	2.1.16 Auffinden des Verzeichnisses von Funktionen	86
	2.2 Vektoren und Matrizen	87
	2.2.1 Teilentnahmen von Elementen bei Vektoren und Matrizen	88
	2.2.2 Automatisierte Bestimmung von Indizes	88
	2.2.3 Automatisierte Bestimmung der Dimensionen	89
	2.2.4 Vorbelegung	90
	2.2.5 Automatisiertes Zusammenfügen von Vektoren und Matrizen	91

2.3	Zeichenketten	92
2.3.1	Grundlagen	92
2.3.2	Klassenumwandlungen	94
2.3.3	Ausführung als MATLAB Anweisung	94
2.4	Structure Array	95
2.5	Cell Array	97
2.6	Objekte	98
2.7	Ablauf- und Kontrollstrukturen	100
2.7.1	If-Verzweigungen	100
2.7.2	Switch-Verzweigung	102
2.7.3	For-Schleife	103
2.7.4	While-Schleife	104
2.7.5	Schleifenunterbrechung (break)	105
2.7.6	Try/catch-Verzweigung	106
2.7.7	Pause	108
2.8	Text einlesen und ausgeben	108
2.9	Daten einlesen und speichern	111
2.9.1	Allgemein übliche Dateiformate	111
2.9.2	MATLAB spezifisches Dateiformat	113
2.10	Grafische Visualisierung	115
2.10.1	Zweidimensionale Visualisierung	116
2.10.2	Dreidimensionale Visualisierung	122
2.11	MATLAB Grundeinstellungen	128
2.11.1	Einrückungen	128
2.11.2	Autosave	129
2.11.3	Kopieren von Grafiken in Dokumente	130
3	Grafische Bedienoberflächen	132
3.1	Grafische Elemente (Graphics Objects)	134
3.1.1	Eigenschaften (Properties)	135
3.1.2	Identifizierungskennzeichen (Handle)	138
3.1.3	Abfrage von Eigenschaften	141
3.1.4	Veränderung von Eigenschaften	144
3.1.5	Hierarchie grafischer Elemente	146
3.1.6	Ermittlung von Identifizierungskennzeichen (Handle)	148
3.1.7	Aktuelles Identifizierungskennzeichen (Handle)	150
3.1.8	Festlegung des Achssystems	151
3.1.9	Achsbeschriftungen	152
3.2	Einführung in die Entwicklung grafischer Bedienoberflächen	153
3.2.1	Anwendungsbeispiel	154
3.2.2	Programmatic GUI	157
3.2.3	Platzierung grafischer Bedienelemente	161
3.2.4	Callback	162
3.2.5	Menüleiste	163
3.2.6	Symbolleiste	166
3.2.7	Ablaufsteuerung	168

3.3	Toolgestützte Entwicklung von grafischen Bedienoberflächen - App Designer	169
3.3.1	Design View	172
3.3.2	Eigenschaften grafischer Bedienelemente	179
3.3.3	Layout der grafischen Bedienoberfläche	183
3.3.4	Code View	185
3.3.5	Properties	188
3.3.6	Nutzung von Callback Functions und Functions in der Bedienoberfläche	190
3.4	Abschließende Bemerkungen	197
3.4.1	Animation	197
3.4.2	Eigenständige Applikationen	197
4	Zahlenformate	199
4.1	Ganze Zahlen	199
4.2	Gleitkommazahlen und Festkommazahlen	206
4.3	Zahlenformate in MATLAB	210
4.4	Über- oder Unterschreitung des Wertebereiches	212
4.5	Auflösungsgrenzen bei Berechnungen	213
4.6	Komplexe Zahlen	215
5	Numerische Integration	216
5.1	Mathematische Problemstellung	217
5.2	Explizites Euler-Verfahren	219
5.3	Runge-Kutta-Verfahren	225
5.4	Berechnungsgenauigkeit und Berechnungsdauer	226
5.5	Einschritt- und Mehrschrittverfahren	228
5.6	Verfahren mit variabler Schrittweite	229
5.7	Steife Systeme	230
5.8	Numerische Integration mit MATLAB	231
6	Zeitgesteuerte Systeme (Simulink)	238
6.1	Modellerstellung	241
6.2	Eigenschaften von Blöcken	258
6.3	Simulation	260
6.4	Visualisierung und Weiterverarbeitung der Simulationsergebnisse ...	264
6.5	Dashboard-Blöcke	269
6.6	Externe Beeinflussung von Blockparametern	274
6.7	Hierarchisches Modell und Verbesserung der Übersichtlichkeit	277
6.8	Model Explorer	282
6.9	Physikalische Modellierung	282
6.10	Codegenerierung	288

7	Ereignisdiskrete Systeme (Stateflow)	289
7.1	Entwicklungsumgebung Stateflow	290
7.2	Beispielsystem	295
7.3	Flussdiagramme	296
7.3.1	Modellerstellung	299
7.3.2	Vorgefertigte Musterabläufe	308
7.3.3	Backtracking	311
7.3.4	Designrichtlinien	313
7.4	Zustandsdiagramme	313
7.4.1	Modellerstellung	316
7.4.2	Aktualisierungsbeispiel	326
7.4.3	Super Step	327
7.4.4	Flussdiagramm in einem Zustand	328
7.4.5	Designrichtlinien	330
7.4.6	Hierarchische Modelle	330
7.4.7	History Junction	334
7.4.8	Parallele Zustände	336
7.4.9	Events	338
7.4.10	Funktionsaufrufe	354
7.5	Tabellarische Beschreibung von Zustandsautomaten	359
7.5.1	Wahrheitstabellen	360
7.5.2	Zustandsübergangstabellen	364
7.6	Simulation und Debugging	371
8	Paralleles Rechnen	374
8.1	Vorarbeit serielle Codeoptimierung	377
8.2	Eingebaute Parallelisierung	378
8.3	Auswahl der Hardware-Ressourcen	380
8.4	Parallele for-Schleifen	382
8.5	Batch jobs und Cluster	384
9	Symbolisches Rechnen	391
9.1	Umformen von algebraischen Ausdrücken	392
9.2	Lösung von Gleichungen	393
9.2.1	Lineare Gleichungen	393
9.2.2	Nichtlineare Gleichungen	395
9.3	Taylorreihen	395
9.4	Laplace-Transformation	396
9.5	Integrieren von Funktionen	396
9.6	Differenzieren von Funktionen	397
9.7	Lösung von Differentialgleichungen	398
	Literatur	402
	Index	403

Hinweis

Folgende Handelsnamen werden häufig verwendet, ohne jedes Mal den Rechteinhaber zu nennen:

- MATLAB® ist eine eingetragene Marke der The MathWorks, Inc.
- Simulink® ist eine eingetragene Marke der The MathWorks, Inc.
- Stateflow® ist eine eingetragene Marke der The MathWorks, Inc.
- Handle Graphics® ist eine eingetragene Marke der The MathWorks, Inc.
- MATLAB® Compiler™ ist eine eingetragene Marke der The MathWorks, Inc.

Für die gezeigten Screenshots und zum Test der aufgeführten Programme bzw. Programm-ausschnitte wurde die MATLAB Revision R2022a verwendet.

Links ins Internet

Zum Teil enthält das Manuskript Informationen, die im Internet zu finden sind. Weiterfüh-
rende Informationen im Internet sind durch folgendes Symbol gekennzeichnet:



Die Informationen waren bei der Ausarbeitung des Manuskriptes im Internet verfügbar. Es kann jedoch nicht gewährleistet werden, dass sie beim Öffnen des Links immer noch vor-
handen sind oder Adressen sich nicht geändert haben.

Formelsymbole

Im gesamten Manuskript wurde versucht, durchgängige und eindeutige Formelsymbole zu verwenden. Bei der ersten Verwendung eines Formelsymbols werden dessen Bezeichnung in Deutsch und Englisch sowie die dazugehörige SI-Einheit und gegebenenfalls wichtige daraus abgeleitete Einheiten angegeben.

F	Kraft	<i>Force</i>	N
M	Drehmoment	<i>Torque</i>	Nm
x	Position	<i>Position</i>	m

Zur Erhöhung der Übersichtlichkeit werden an manchen Stellen diese Angaben wiederholt.

Programmbeispiele

Die mit Icons gekennzeichneten Programmbeispiele sind kapitelweise unter *plus.hanser-fachbuch.de* abgelegt.

Icon	Files	
	MATLAB	Script
		Function
		Figure
		App
		MAT-file
		MEX-file
	Simulink und Stateflow	Model
	Project	

Von dort können diese frei für Ausbildungszwecke, die nicht kostenpflichtig sind, heruntergeladen werden.

1

Einführung

Die Arbeit von Ingenieuren bei der Produktentwicklung erfolgt immer stärker computerunterstützt. Grund hierfür ist, dass Computer bei gleichen oder sinkenden Kosten kontinuierlich leistungsfähiger werden. Selbst von der erforderlichen Computerleistung sehr anspruchsvolle Aufgaben können immer mehr am Arbeitsplatzrechner eines Ingenieurs gelöst werden. Programme (Software), die die Arbeit von Ingenieuren unterstützen, werden unter dem englischen Begriff „Computer Aided Engineering“, oder abgekürzt CAE, zusammengefasst. Sowohl bei der Entwicklung mechanischer als auch elektronischer Baugruppen gibt es eine Vielzahl von Programmen, die einzelne Aufgaben bei der Produktentwicklung unterstützen (Bild 1.1).

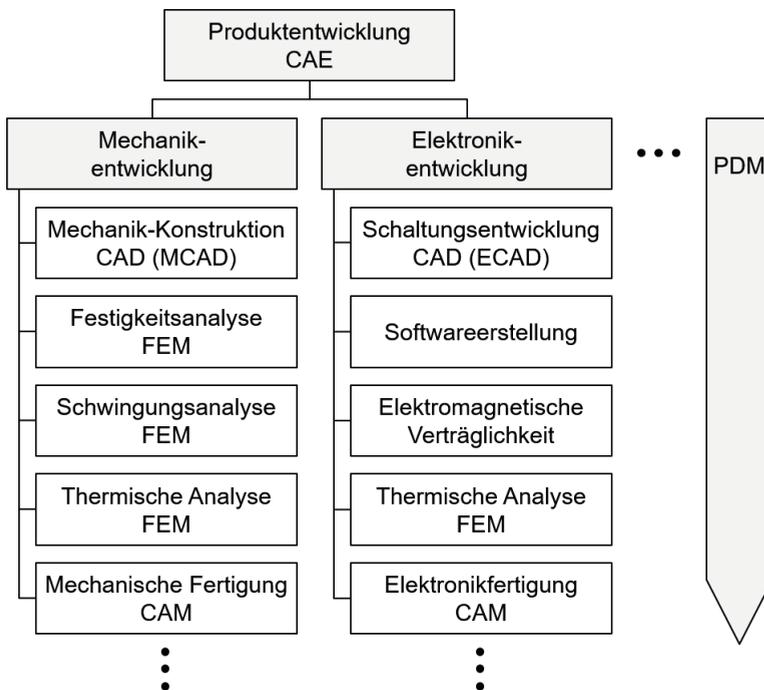


Bild 1.1 Computerunterstützung bei der Produktentwicklung (Beispiele)

Die Konstruktion mechanischer Teile und Baugruppen erfolgt fast ausschließlich mit CAD-Systemen (CAD: Computer Aided Design). In Bild 1.2 ist die Veränderung der Arbeitsweise in der mechanischen Konstruktion vom Arbeiten am Zeichenbrett mit Bleistift und Tusche hin zum Arbeiten am Computer mit dreidimensionaler Darstellung gezeigt. Die Berech-

nung der mechanischen Festigkeit, der Verformung und des Schwingungsverhaltens erfolgt mit Software, die auf speziellen mathematischen Verfahren basiert. Diese Verfahren sind unter dem Begriff FEM (Finite-Elemente-Methode) zusammengefasst. Weitere Anwendungen im Ingenieurbereich sind z. B. thermische Analysen, Strömungssimulationen oder die Berechnung von elektrischen und magnetischen Feldern. Einige Anwendungsbeispiele zeigen Bild 1.3, Bild 1.4 und Bild 1.5.

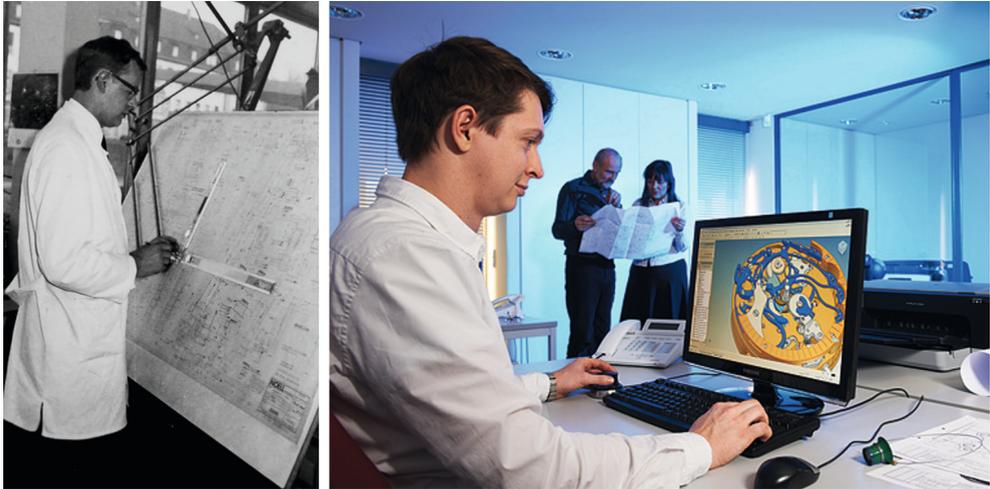


Bild 1.2 Mechanische Konstruktion. Linkes Bild: Konstruktion am Zeichenbrett (© Ing. B. P. Hennek, Würzburg 1968), rechtes Bild: Konstruktion mit CAD-System (© Tutima Uhrenfabrik GmbH Ndl. Glashütte, 2016)

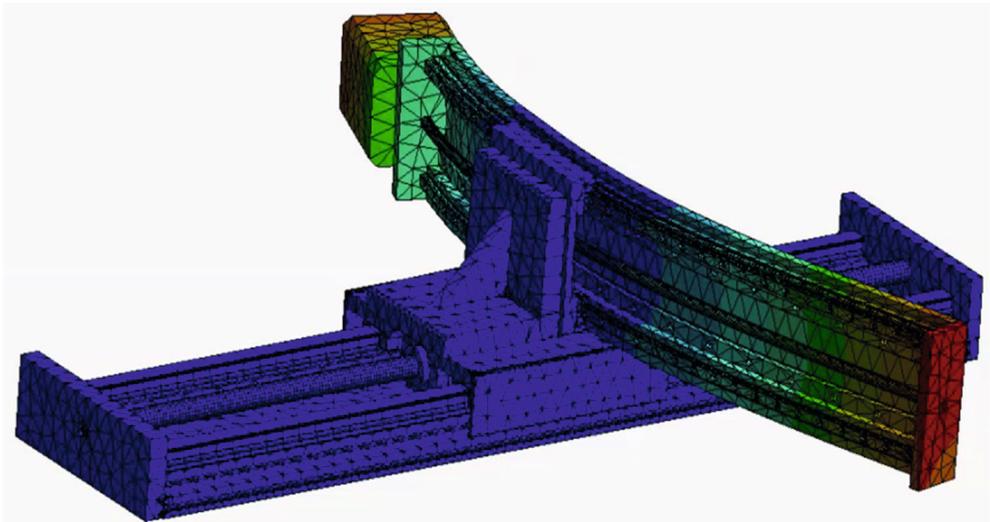


Bild 1.3 Anwendungsbeispiele Finite-Elemente-Berechnung – Verformungsanalyse XY-Tisch (© Labor für elektrische Antriebstechnik, Hochschule Rosenheim, 2016)

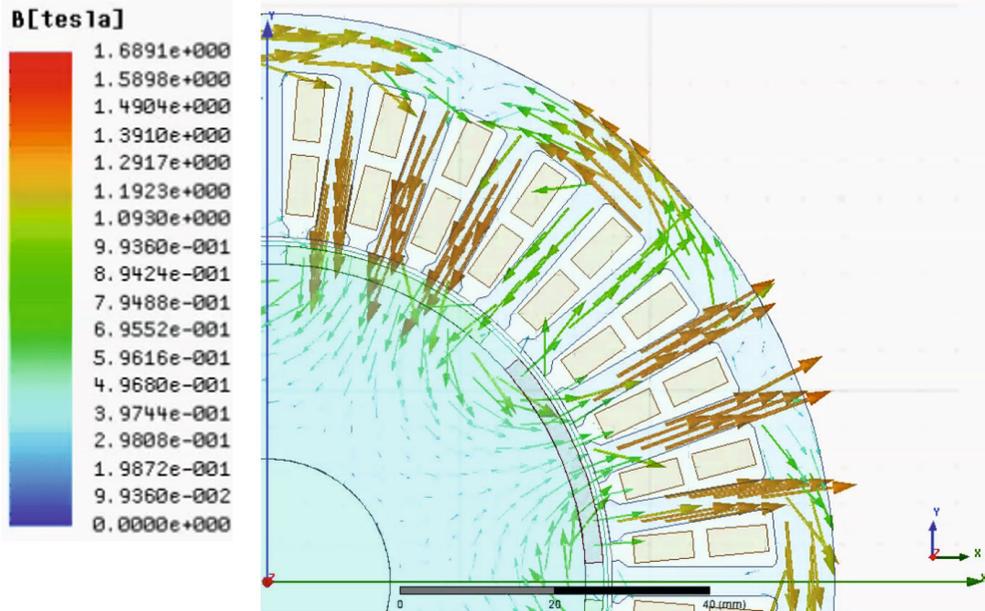


Bild 1.4 Anwendungsbeispiele Finite-Elemente-Berechnung – Magnetfeldberechnung eines Elektromotors (© Labor für elektrische Antriebstechnik, Hochschule Rosenheim, 2016)

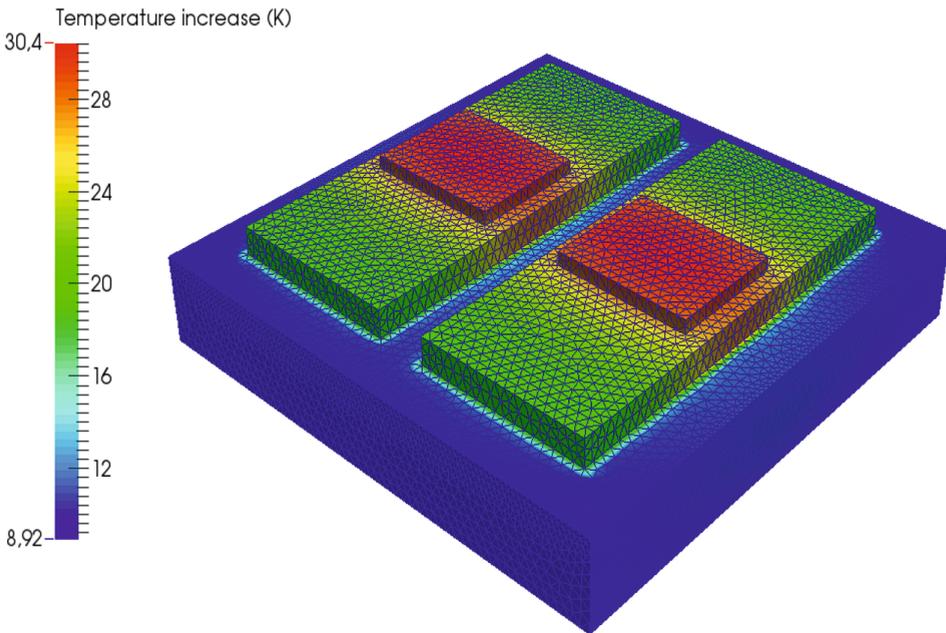


Bild 1.5 Anwendungsbeispiele Finite-Elemente-Berechnung – Temperaturverteilung von Halbleiterchips auf einem Schaltungsträger (© Prof. Dr. techn. Norbert Seliger, Labor für Leistungselektronik, Hochschule Rosenheim, 2016)

Um alle Dokumente und Daten der Produktentwicklung in einem System zu speichern und zu verwalten, gibt es Software zum Produktdatenmanagement (PDM). Neben der Entwicklung speichern dort auch andere Unternehmensbereiche, wie Produktion, Service oder Vertrieb, produktspezifische Informationen ab. Alle für ein Produkt relevanten Informationen werden dort archiviert.

Einige Teilbereiche der computerunterstützten Entwicklung und dazugehörige Produktbeispiele zeigt Tabelle 1.1.

Tabelle 1.1 CAE-Bereiche und Beispielprodukte für den jeweiligen Bereich

Bereich	Software	
Computer Aided Design CAD		
Mechanisch	M-CAD	CATIA®, NX™, Creo Parametric®, Solid Edge®, SolidWorks®,...
Elektronisch	E-CAD	cadence®, SYNOPSIS®,...
Computer Aided Manufacturing CAM		
Mechanisch	M-CAM	hyperMILL®, NX™, PowerMILL®, Tebis™,...
Elektronisch	E-CAM	
Ingenieurmathematik		
Finite Element Berechnung (Finite Element Methode)	FEM	ABAQUS®, ANSYS®, LS-DYNA®, NASTRAN®, PERMAS®,...
Lösung mathematischer Aufgaben		GNU Octave, SCILab, Maple®, Mathematica®, MATLAB®,...
Regelungs- und Steuerungseinrichtungen		
Modellbasierte Entwicklung		Simulink®, Stateflow®

- CATIA®, SolidWorks® sind eingetragene Marken der Dassault Systèmes SE
- NX™, Solid Edge® sind Marken der Siemens Product Lifecycle Management Software, Inc.
- Creo Parametric® ist eine eingetragene Marke der PTC, Inc.
- cadence® ist eine eingetragene Marke der Cadence Design Systems, Inc.
- SYNOPSIS® ist eine eingetragene Marke der Synopsys, Inc.
- hyperMILL® ist eine eingetragene Marke der OPEN MIND Technologies AG
- PowerMILL® ist eine eingetragene Marke der Delcam Ltd
- tebis™ ist eine eingetragene Marke der Tebis Technische Informationssysteme AG
- ABAQUS® ist eine eingetragene Marke von Abaqus, Inc.
- ANSYS® ist eine eingetragene Marke von Ansys, Inc.
- LS-DYNA® ist eine eingetragene Marke von Livermore Software Technology Corp.
- NASTRAN® ist eine eingetragene Marke der National Aeronautics Space Administration. MSC Nastran ist eine häufig eingesetzte Version die von der MSC Software Corporation entwickelt und gewartet wird.
- PERMAS® ist eine eingetragene Marke der Intes GmbH
- Maple™ ist eine Marken der Waterloo Maple, Inc.
- Mathematica® ist eine eingetragene Marke der Wolfram Research, Inc.
- MATLAB®, Simulink®, Stateflow® sind eingetragene Marken der The MathWorks, Inc.

■ 1.1 Historie Rechenmaschinen

Für sehr viele Produkte werden heute computerbasierte Systeme eingesetzt. Viele Produkte sind durch die Entwicklung leistungsfähiger, miniaturisierter, kostengünstiger und verbrauchsarmer Computer erst möglich geworden. Hierzu zählen:

- Personal Computer, für den einzelnen Büroarbeitsplatz, den Privatgebrauch und den mobilen Einsatz (Notebooks, Tablet, ...)
- Smartphones
- Navigationssysteme

- Antilockersysteme und Stabilisierungssysteme
- Spielkonsolen
- Flexible und hochautomatisierte Produktionseinrichtungen, wie Industrieroboter, Werkzeugmaschinen, Verpackungsmaschinen etc.

Die Entwicklung hin zu heutigen Computersystemen hat eine lange Historie. Erste Rechenmaschinen zur Lösung einfacher Berechnungsaufgaben wurden mechanisch aufgebaut. Die erste bekannte Rechenmaschine, die alle vier Grundrechenarten (Addition, Subtraktion, Multiplikation und Division) ausführen konnte, war die Leibniz'sche Rechenmaschine (Bild 1.6). Basis für die ersten teilweise elektrischen Rechenmaschinen waren Telefonrelais in Kombination mit Schrittschaltwerken (Bild 1.7). Mit ca. 2000 Telefonrelais wurde der erste, voll funktionsfähige, programmierbare Computer der Welt (Zuse Z3) aufgebaut (Bild 1.8). Einige technische Daten dieses Computers zeigt Tabelle 1.2.

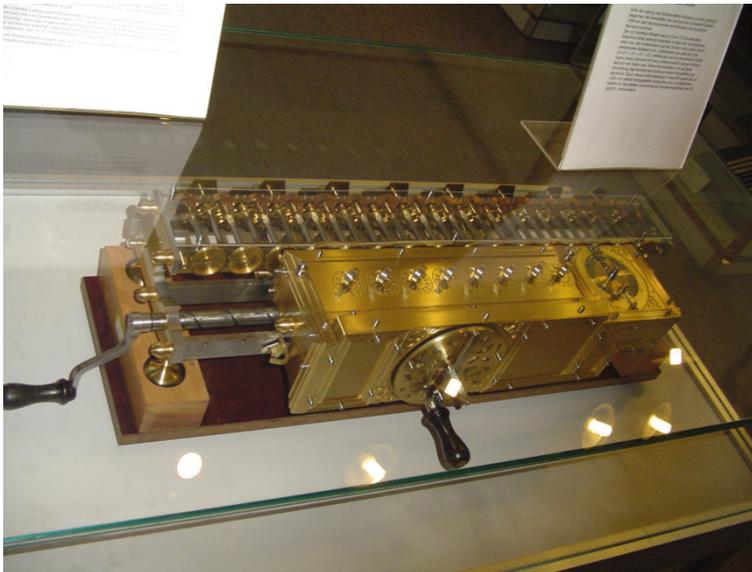


Bild 1.6 Mechanische Rechenmaschine (Leibniz'sche Rechenmaschine), entwickelt von Wilhelm Leibniz ca. 1700 in Dresden (© Kolossos, Technische Sammlungen der Stadt Dresden, Wikipedia, 2016) <https://upload.wikimedia.org/wikipedia/commons/9/92/Leibnitzrechenmaschine.jpg>

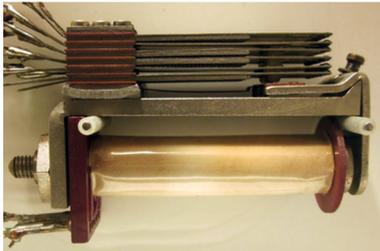
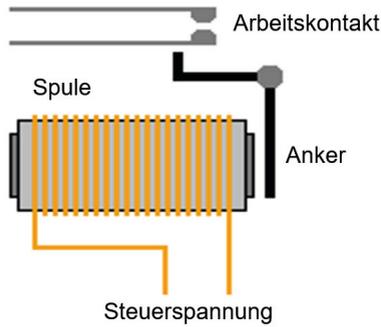


Bild 1.7 Oben: Relaisprinzip und Relais aus Zuse Z3 (© oben: Nogo, Wikipedia, 2016; unten: Denis Apel, Wikipedia, 2016)

https://commons.wikimedia.org/wiki/File:Relais_ruhe.png#/media/File:Relais_ruhe.png

https://de.wikipedia.org/wiki/Zuse_Z3#/media/File:Elektromagnetischerspeicher_zuse_relais.jpg

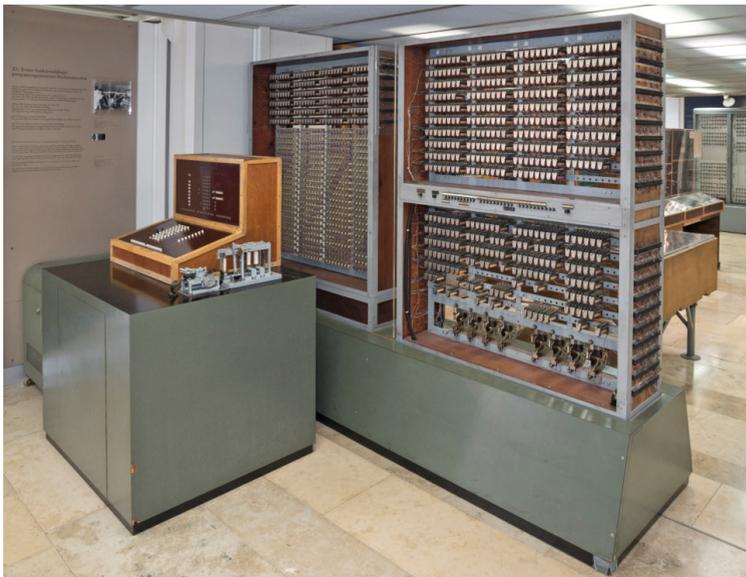


Bild 1.8 Elektromechanische Rechenmaschine (Zuse Z3), entwickelt von Konrad Zuse 1941 in Berlin (© Archiv Deutsches Museum, München, 2016) <http://www.deutsches-museum.de/archiv>

Index

Symbole

.mlapp 172

A

Ablaufsteuerung 168
Ablaufstrukturen 100
Achsbeschriftung 152
Achssystem 151
Achssystem, Festlegung 151
Action 314
Action table 361
Adams-Bashforth-Verfahren 228
analytische Lösung 221
Anfangswert 217
Anfangswertproblem 217
Animation 197, 372
Anonymous Function 72
Anweisung 318
App Designer 169
Arbeitsverzeichnis 77
arithmetische Operatoren
 (Arithmetic operators) 51
ASCII 93
Assemblersprache 37
Attribut 185
Auflösung 213
Ausgabeargumente 66
Ausgabengrößen 66
automatische Codegenerierung 31
Axes 138, 159, 175
Axes objects 134, 157

B

Base Workspace 75
Batch jobs 384
Bedienoberfläche, grafische 132

bedingte Aktion 296, 302
Bedingung 296, 302
benannte Felder (named fields) 95
Benutzerschnittstelle 132
Berechnungsfehler 221, 226
Berechnungsgenauigkeit 216, 227
Berechnungszeit 227
Binärzahlen 199
biologische Systeme 238
black box 66
Blockorientierte Modellierung 238
Breakpoint 372
broadcast 338
Browser-basierte Hilfe 47
built-in functions 66
Button 174
Button group 159

C

Callback 162, 168, 190, 195, 259, 307
Callback Function 259
Canvas 173
case 102
catch 106
cd 78
cell 97
cell2mat 97
Cell Array 97 f.
char 93
character 43
Chart 292, 316
Chart-level 320
Chart Properties 298
Check Box 158, 174
children 147
Class (Value) 43
clear 84

- Cluster 384
- Code View 170, 185
- coeffs 392
- Command Window 41, 133
- Comment (Kommentar) 324
- Compiler 37
- Component Browser 176
- Component Library 173
- Computeralgebrasystem 29
- Condition 296, 302, 324, 365
- Condition Action 302, 324, 365
- Condition table 361
- Configuration Parameters 260
- Connective junctions 296
- Constant-Block 256
- Core 374
- createJob 386
- createTask 386
- Current directory 77
- Current folder 77
- Current working folder 77

D

- Dashboard 335
- Datentyp (Wert) 43
- Debugger 372
- Debug-Modus 83
- Default Transition 296, 313, 366
- delete 387
- Design View 170 ff.
- Dezimalzahlen 41, 199
- diag 90
- diff 397
- Dimension 89
- diskrete Elektronik 25
- Diskretisierungsfehler 226
- disp 109
- Distributed computing 375
- Divisionsmethode 200
- Dokumentation 46
- double 210
- dreidimensionale Visualisierung 122
- Drop Down 175
- dsolve 398

E

- Edit box 158
- Edit Field 174
- Eigenfrequenz 156
- Eigenkreisfrequenz 155
- Eigenschaften 233, 258
- Eigenschaften, Abfrage 141
- Eigenschaften, Veränderung 144
- eigenständige Applikation 197
- eindimensionale Felder 49
- Eingabeargumente 66
- Eingabedefinition (Notation) 42
- Eingabegrößen 66
- elektronische Steuergeräte 31
- else 101
- elseif 101
- elseif-Verzweigung 101
- Embedded Coder 288
- end 91
- Endlosschleife 105
- ENIAC 23
- ereignisdiskrete Systeme 31
- eval 94 f.
- evalc 95
- evalin 95
- Event (Ereignis) 324, 338
- Event Name 339
- Execution Order 320, 337
- Exklusive Zustände 291, 336
- expand 393
- eye 90

F

- false 43
- fetchOutputs 387
- fields 98
- figure 134, 138
- Figure object 134, 157
- Finanzsysteme 238
- find 88
- findobj 149
- Flip Block 250
- Flussdiagramm 31, 294 ff.
- Formatierungsoperatoren 110
- For-Schleife 103

Foundation Library 283
 fprintf 109
 function 57, 66 f., 190 f.
 Function-Call-Subsystems 348
 Function Functions 71
 Function Handle 71, 233, 386
 Function Workspace 75
 Funktionsauswertung 220

G

ganze Zahlen 41, 199
 gca (get current axes) 150
 (gcf (get current figure) 150
 gcp 381
 geschlossene Lösung 28 f.
 get 142, 148
 Gleitkommazahlen 41
 globaler Fehler 226
 Global Function 73
 Global Variables 75
 grafische Bedienelemente, Platzierung 161
 grafische Bedienoberfläche, Entwicklung 153
 grafische Eigenschaften 135
 grafische Elemente 134
 grafische Elemente, Hierarchie 146
 grafische Visualisierung 115
 Graphical User Interface (GUI) 133
 Graphics Objects 134
 grid 152
 GUIDE 170

H

Handle 138, 148 ff.
 Handle Graphics® 134
 Hardware-in-the-Loop 33
 Help 47
 Help Browser 48
 Hierarchie 147
 High Byte 203
 Hilfeunterstützung 46
 History Junction 334
 Hochsprachen 37

hold off 119
 hold on 119

I

Identifizierungskennzeichen 138, 148 ff.
 If-Verzweigungen 100
 importdata 113
 Inf 52
 Inherit 306
 Initial condition 256
 Inline Functions 75
 input 108
 int 392
 int2str 94
 int8 210
 int16 210
 int32 210
 int64 210
 integer numbers 199
 integrierte Schaltkreise 25
 Interpreter 37
 intmax 211
 intmin 211
 isempty 76

J

Job 386

K

Klassenumwandlung 94
 Knotenpunkte 294 ff.
 komplexe Zahlen 41, 199, 215
 kontextsensitive Menüleiste 165
 Kontrollstrukturen 100

L

Label 175
 Labs 377
 laplace 396
 LaTeX-Notation 119
 Layout 183

Least Significant Bit 203
Leibniz'sche Rechenmaschine 21
length 89
Limits 179f.
Line object 134f.
LineSpec 117
Linienart 117
Linienfarbe 117
List box 159
load 115
local functions 73
logical values 43
logische Operatoren 53
logische Verzweigungen 100
Log Selected Signal 267
lokaler Fehler 226
Lösungsalgorithmus solver 234
Lösungsverfahren 223, 231 ff.
Low Byte 203

M

main function 72f.
Maple® 29
Markierungen 117
Maschinensprache 37
Masse 217
mat 113
Mathematica® 29
MATLAB Editor 57
MATLAB Grafikfenster 116
MATLAB Interpreter 94
MATLAB Lizenznummer 85
MATLAB Scheduler 385
MATLAB Syntax 94
MATLAB Version 85
MATLAB Versionsnummer 85
Matrix 50
Matrizen 87
mehrdimensionale Felder 49
Mensch-Maschine-Schnittstelle 132
Menüleiste 163
meshgrid 125
Method 185
Mikroprozessor 25
Mittelpunktsregel 228

Model Browser 280
Model Explorer 306, 340, 351, 354, 362
Model Hierarchy 306
modellbasierte Entwicklung 31
Moore'sches Gesetz 27
Most Significant Bit 203
Multithreaded functions 378
Musterabläufe 308

N

Näherungslösung 216
Name 43
NaN 53
negative Zahlen 204
Nested Functions 75
New Script 57
normalized 161
num2str 94
numerische Integration 231
numerische Lösung 29
numerische Lösungsverfahren 241
numerische Mathematik 28
numerisches Berechnungsverfahren 28
numerische Verfahren 216

O

Objekt 98
objektorientierte Programmierung
(Object-oriented Programming) 98
Odefun 233
ones 90
OpenFcn 259
Opening Function 259
Options 233
otherwise 102

P

Panel 159, 175
Parallel batch jobs 384
Parallel computing 374
Parallele Zustände 291, 336
Parallel pool 380
parcluster 385

parents 147
 parpool 380
 path 78
 Pattern Wizard 308
 pause 108, 197
 Periodendauer 156
 Persistent Variables 76
 Physikalische Modellierung
 (objektorientierte Modellierung) 238
 plot 116, 138
 plot3 122
 Pop-up menu 159
 Port 306
 Present working directory 77
 pretty 392
 Private Functions 75
 Programmatic GUI 157
 Programmiersprache 37
 Properties 135, 233, 258
 Property 179, 185, 188
 Property Inspector 136, 144
 Prozess 376
 PS-Simulink Converter 284
 Pushbutton 157, 166
 pwd 78

R

Radio button 158
 Rapid Control Prototyping 33
 realmax 211
 realmin 211
 Recheneinheiten 374
 reelle Zahlen 199
 relationale Operatoren 53
 Revision 85
 Rotate Block 250
 Rückverfolgung (Backtracking) 311
 Rundungsfehler 226
 Runge-Kutta-Verfahren 225

S

save 114
 Schaltsignal 289
 Schleifen 100

Schleifenunterbrechung (break) 105
 Schlüsselwort (Keyword) 318
 Schrittweite 221, 226 ff.
 Scope 265, 354
 Script 57, 66
 Search Path 77
 Separator 168
 serielles Rechnen 374
 set 144
 Signalleitungen 248
 simplify 393
 Simscape 282
 Simulation 260, 371
 Simulation Data Inspector 267
 Simulink Coder 288
 Simulink Library Browser 269, 292, 316
 Simulink PS Converter 284
 Simulink Start Page 293
 single 210
 Sinks 252
 size 89
 Slider 158
 solve 393
 solver 231 ff.
 Sonderzeichen 55
 Spaltenvektor 50
 Special character 56
 Speicherbedarf 211
 Spinner 175, 180
 Stammbaum 147
 Startzustand (Default State) 314
 State 314
 State Action 317 ff.
 State chart 313
 Stateflow 290 ff.
 Stateflow Debugger 372
 Stateflow Editor 292, 300, 316
 Stateflow Layoutbereich 294
 State Label 317
 State-level 320
 Statement 318
 State name 317
 State Transition Matrix 367
 State Transition Table 359, 364
 Static text 158
 strcmp 92

strfind 92
string 43, 92
struct 95
Structure Array 95, 98
Subfunctions 73
subplot 120
subs 392
Substate 330, 369
Subtraktionsmethode 200
Superstate 330, 369
Super Step 327
surf 122 f.
switch 102
Switch-Verzweigung 102
symbolische Lösung 29
symbolisches Lösungsverfahren 29
symbolisches Objekt 391
symbolisches Rechnen 391
Symbolleiste 166
Syntaxdefinition 55

T

Table 159
Task 385
taylor 395
technische Systeme 238
Terminating Transition 312
termination junction 296
Textzeichen 43
Thread 376
tic 377
title 152
toc 377
Toggle Button 159, 166
Toolbox 32
Tooltip 179 ff.
To Workspace 252
TRADIC 24
Transistor 24
Transition 294 ff., 314, 365
Transition action (Übergangsaktion)
324
Transition Label 296, 302, 319
Triggered Subsystems 348
true 43

Truth Table 359 f.
try 106

U

Übergangsbedingungen 296
uicontextmenu 161, 165
UIcontrol objects 157
uimenu 161 ff.
uint8 210
uint16 210
uint32 210
uint64 210
uipanel 161
uipushtool 161
uitable 161
uitoggletool 161
uitoolbar 161, 166
Unicode® 93
Units 161
Unterbrechungen 100
Unterprogramm 66

V

Variablendefinition 41 ff.
Variablenname 43
Variablenzuweisung 43
variable Schrittweite 229
Variables Editor 87, 133
Vektoren 87
ver 85
Visualisierung 64
Vorbelegung 90
Vorlage (Template) 67
Vorzeichenbit 204

W

Wahrheitstabelle 294
Wahrheitswert 43, 289, 303
wait 387
Wertebereich 199, 212
wertediskret 289
Wertezuweisung 41
While-Schleife 104

Worker 377, 385
Workspace 41
Workspace Browser 41, 140

X

xlabel 152
XY Graph 252

Y

ylabel 152

Z

Zahlenauflösung 214
Zahlendarstellung 199
Zeichenketten 43, 92
Zeilenvektor 50
zeitdiskret 289
zeitgesteuerte Systeme 31
zeros 69, 90
Zuse Z3 21
Zustand 294, 317, 365
Zustandsautomaten 290, 313
Zustandsdiagramme 294, 313
Zustandsgraph 31
Zustandsgröße 218
Zustandsübergangstabelle 294, 364
zweidimensionale Visualisierung 116
Zweierkomplement 204