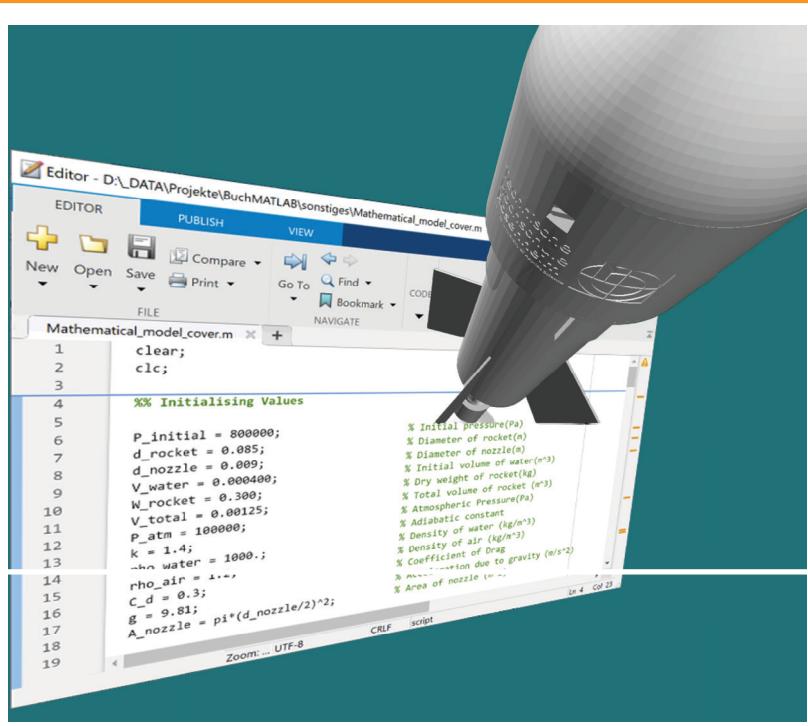


Rainer Hagl
Frank King
Peter Zentgraf



Ingenieurinformatik

Eine Einführung mit MATLAB,
Simulink und Stateflow



2., überarbeitete Auflage

HANSER



Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial. Geben Sie dazu einfach diesen Code ein:

plus-q2wbk-hhw27

plus.hanser-fachbuch.de



Bleiben Sie auf dem Laufenden!

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

www.hanser-fachbuch.de/newsletter

Rainer Hagl/Frank A. King/Peter Zentgraf

Ingenieurinformatik

Eine Einführung mit MATLAB[®], Simulink[®] und Stateflow[®]

2., neu bearbeitete Auflage

HANSER

Die Autoren:

Prof. Dr.-Ing. Rainer Hagl, Technische Hochschule Rosenheim

Prof. Dr.-Ing. Frank A. King, Technische Hochschule Rosenheim

Prof. Dr.-Ing. Peter Zentgraf, Technische Hochschule Rosenheim



Alle in diesem Buch enthaltenen Informationen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt geprüft und getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor(en, Herausgeber) und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso wenig übernehmen Autor(en, Herausgeber) und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürfen.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, sind vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2023 Carl Hanser Verlag München

Internet: www.hanser-fachbuch.de

Lektorat: Frank Katzenmayer

Herstellung: Frauke Schafft

Covergestaltung: Max Kostopoulos

Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München

Titelbild: © Frank A. King/Peter Zentgraf

Satz: Eberl & Koesel Studio, Kempten

Druck und Bindung: CPI books GmbH, Leck

Printed in Germany

Print-ISBN 978-3-446-47515-1

E-Book-ISBN 978-3-446-47730-8

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	Wertzuweisung 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

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. Weiterführende 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 vorhanden 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).

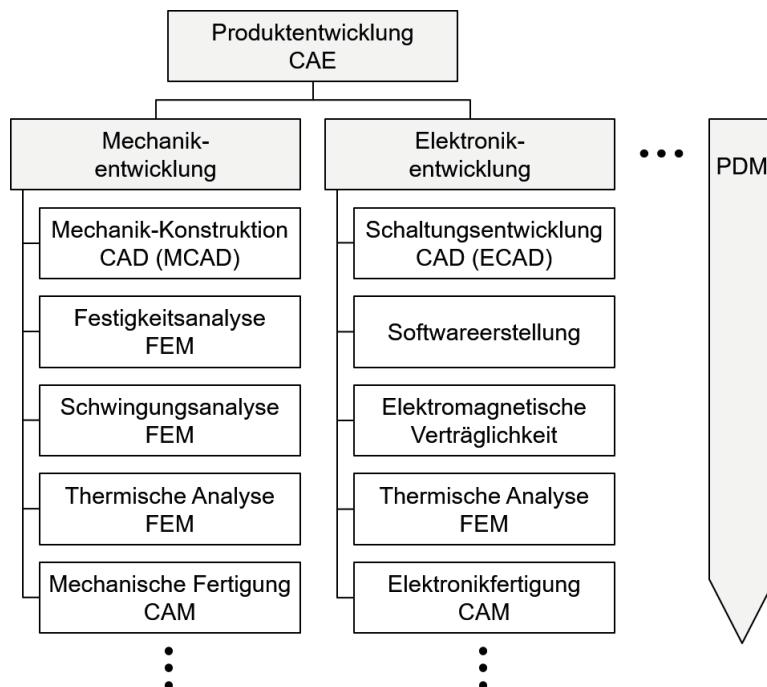


Bild 1.1 Computerunterstützung bei der Produktentwicklung (Beispiele)

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.

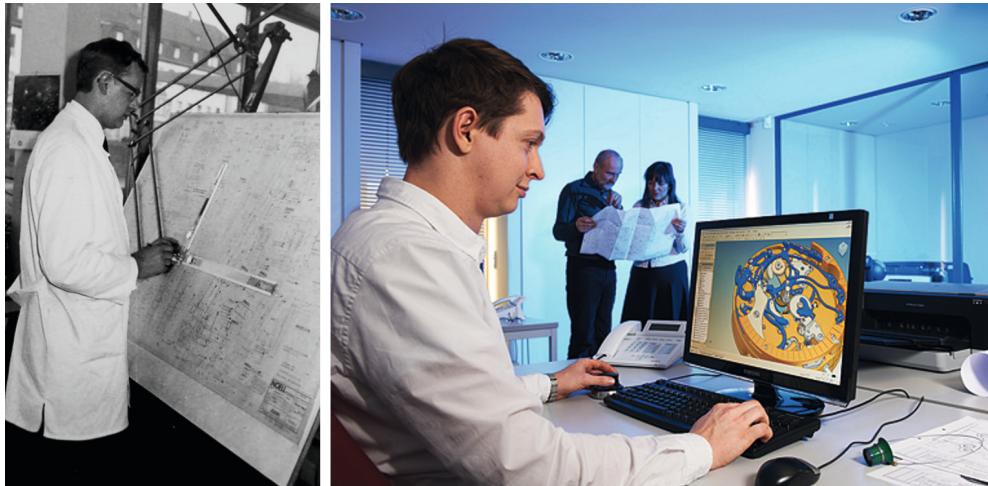


Bild 1.2 Mechanische Konstruktion. Linkes Bild: Konstruktion am Zeichenbrett (© Ing. B. P. Hennek, Würzburg 1968), rechtes Bild: Konstruktion mit CAD-System (© Tutima Uhrenfabrik GmbH Ndl. Glashütte, 2016)

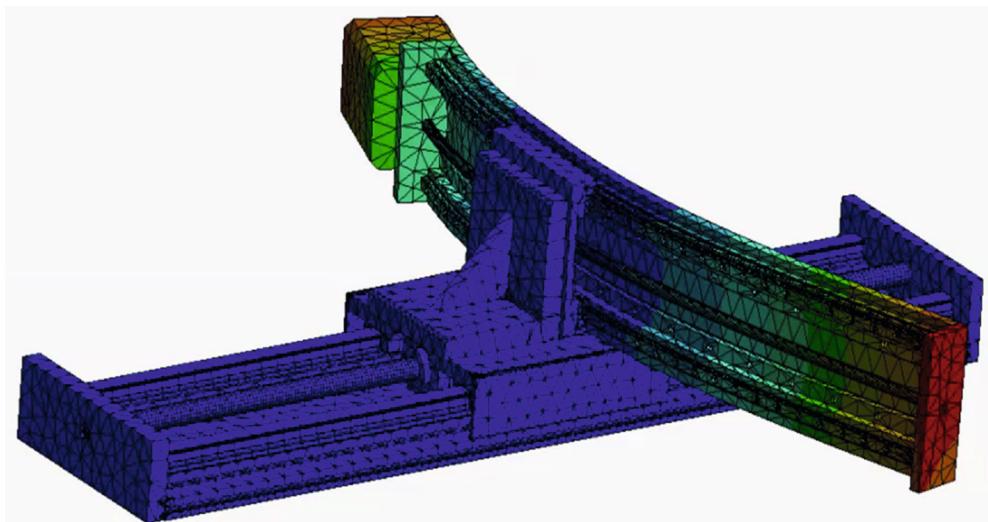


Bild 1.3 Anwendungsbeispiele Finite-Elemente-Berechnung – Verformungsanalyse XY-Tisch (© Labor für elektrische Antriebstechnik, Hochschule Rosenheim, 2016)

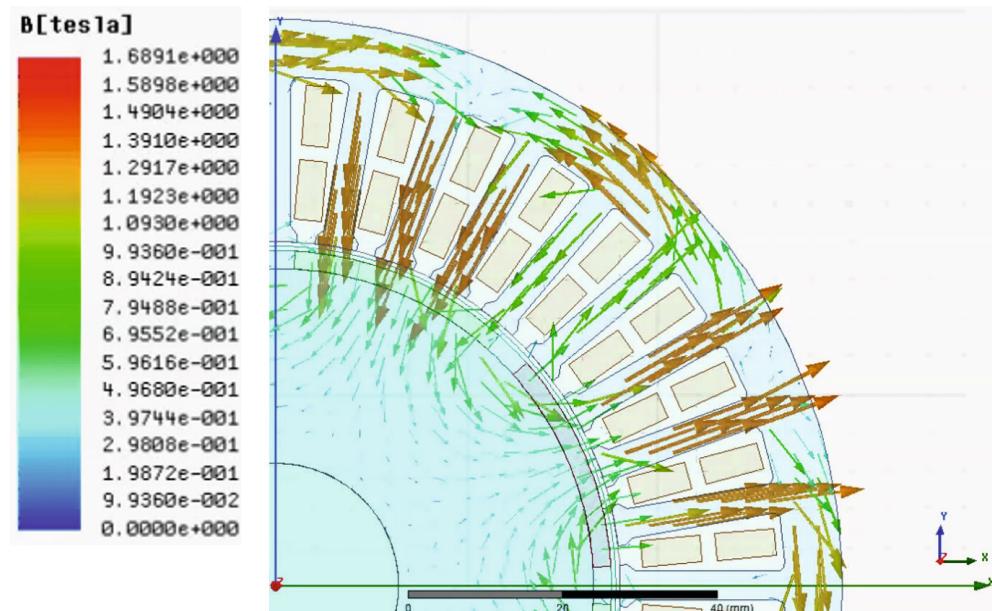


Bild 1.4 Anwendungsbeispiele Finite-Elemente-Berechnung – Magnetfeldberechnung eines Elektromotors (© Labor für elektrische Antriebstechnik, Hochschule Rosenheim, 2016)

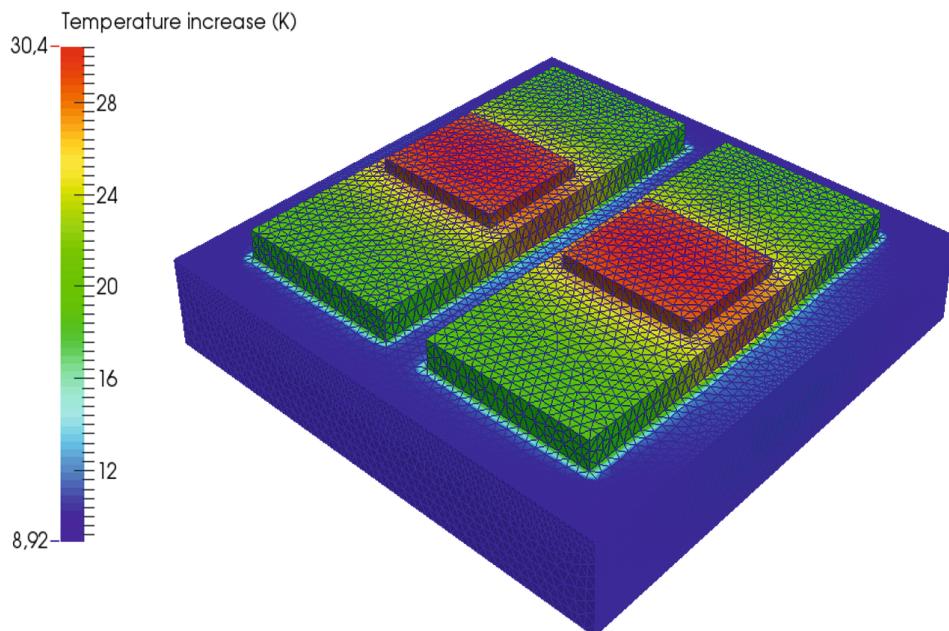


Bild 1.5 Anwendungsbeispiele Finite-Elemente-Berechnung – Temperaturverteilung von Halbleiterchips auf einem Schaltungsträger (© Prof. Dr. techn. Norbert Seliger, Labor für Leistungselektronik, Hochschule Rosenheim, 2016)

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®, SYNOPSYS®,...
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®
<ul style="list-style-type: none"> • 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. • SYNOPSYS® 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 Marke 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

- Antiblockiersysteme 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.

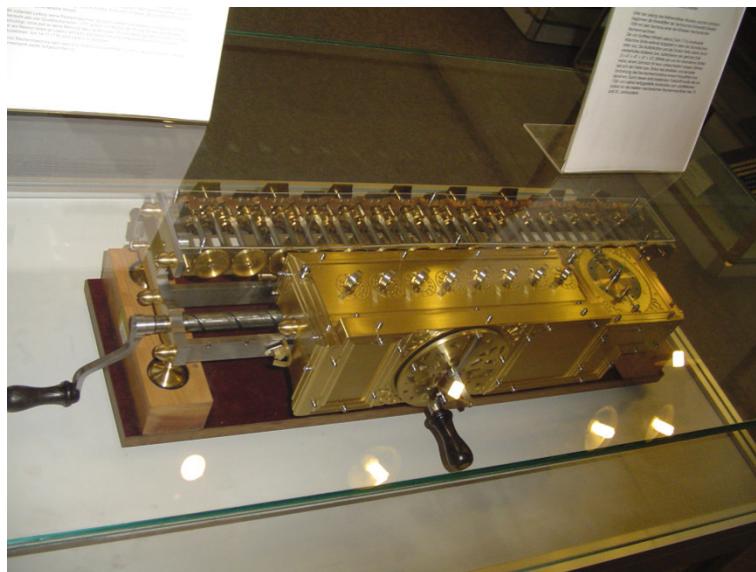


Bild 1.6 Mechanische Rechenmaschine (Leibniz'sche Rechenmaschine), entwickelt von Wilhelm Leibniz ca. 1700 in Dresden (© Kolossos, Technische Sammlungen der Stadt Dresden, Wikipedia, 2016) <https://upload.wikimedia.org/wikipedia/commons/9/92/Leibnitzrechenmaschine.jpg>

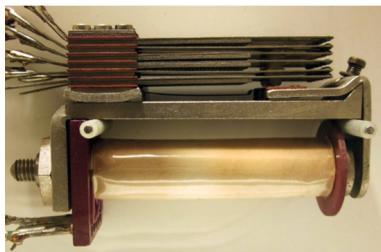
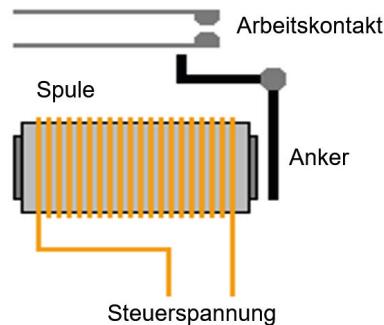


Bild 1.7 Oben: Relaisprinzip und Relais aus Zuse Z3 (© oben: Nogo, Wikipedia, 2016; unten: Denis Apel, Wikipedia, 2016)

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

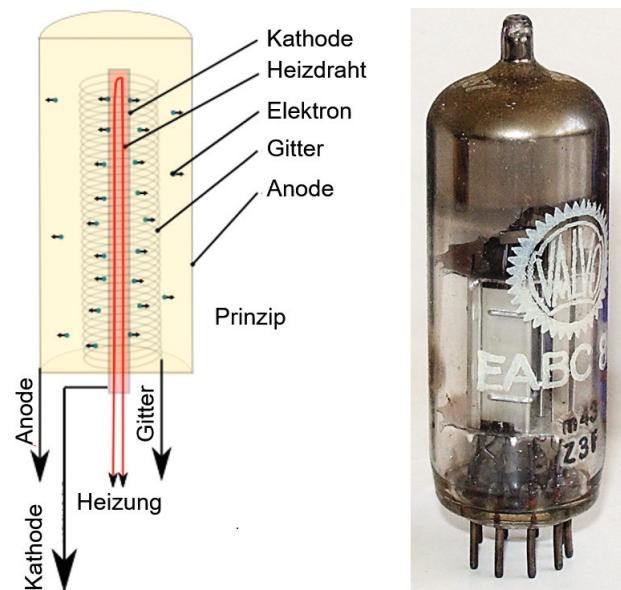


Bild 1.8 Elektromechanische Rechenmaschine (Zuse Z3), entwickelt von Konrad Zuse 1941 in Berlin (© Archiv Deutsches Museum, München, 2016) <http://www.deutsches-museum.de/archiv>

Tabelle 1.2 Technische Daten des ersten Computers (ZUSE Z3)

Taktfrequenz	ca. 5,4 Hz
Rechenwerk (2 Register)	600 Relais
Speicher (64×22 bit)	1400 Relais
Fließkommaeinheit	22 bit
Aufgenommene Leistung	4 kW
Addition	3 Takte
Multiplikation	16 Takte
Division	28 Takte
Ein-/Ausgabe-Einheit	Lochkartenleser für handgestanzten 35 mm breiten Film

Die elektromechanischen Lösungen hatten keine hohe Verfügbarkeit und waren mit hohem Wartungsaufwand verbunden. Sie wurden von rein elektrischen Systemen, die zunächst Elektronenröhren (Bild 1.9) nutzten, abgelöst. Der erste elektrische Universalrechner (ENIAC) wurde an der University of Pennsylvania im Auftrag der US-Armee von 1942 bis 1946 entwickelt (Bild 1.10).

**Bild 1.9** Elektronenröhre (© links: Stefan Riepl, Wikipedia, 2016; rechts: Ausschnitt 32-bit-Maschine, Wikipedia, 2016)

https://de.wikipedia.org/wiki/Elektronenr%C3%B6hre#/media/File:Elektronenroehre_real.png
https://de.wikipedia.org/wiki/Elektronenr%C3%B6hre#/media/File:Radio_vacuum_tubes.jpg

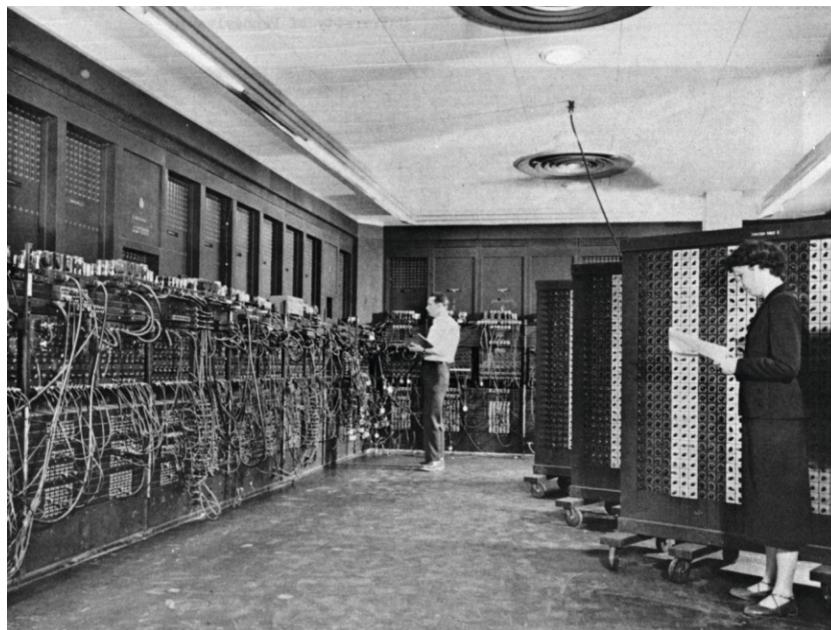


Bild 1.10 Elektrische Rechenmaschine mit Elektronenröhren, Electronic Numerical Integrator and Computer (ENIAC) (© U.S. Army, Wikipedia, 2016) <https://de.wikipedia.org/wiki/ENIAC#/media/File:Eniac.jpg>

Durch die Entwicklung von Halbleitern und Leiterplatten konnten elektronische Systeme stark verkleinert und in der Leistungsaufnahme reduziert werden. Insbesondere die Halbleiterbauelemente Transistor (Bild 1.11) und Diode ermöglichen, im Vergleich zu Lösungen auf Basis von Elektronenröhren, kompaktere, wartungsärmere und leistungsfähigere Rechenmaschinen. Die erste vollständig elektronische Rechenmaschine wurde von den Bell-Forschungslaboratorien für die U.S. Air Force von 1953 bis 1955 (TRADIC) entwickelt (Bild 1.12).

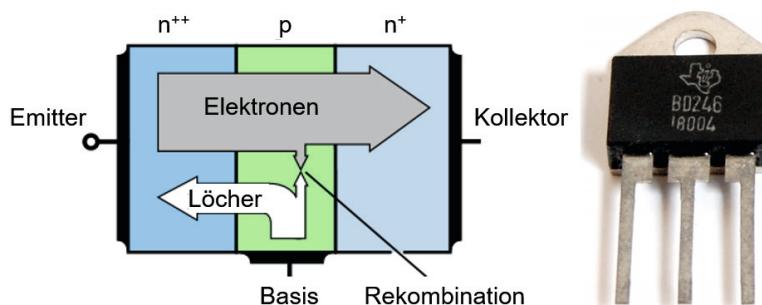


Bild 1.11 Links: Funktionsprinzip Transistor, rechts: Bauelement (© links: Kai Martin, Cepheiden, Wikipedia, 2016; rechts: Ausschnitt Benedikt Seidl, Wikipedia, 2016) https://de.wikipedia.org/wiki/Transistor#/media/File:NPN_transistor_basic_operation.svg <https://upload.wikimedia.org/wikipedia/commons/0/0e/Transistors-white.jpg>

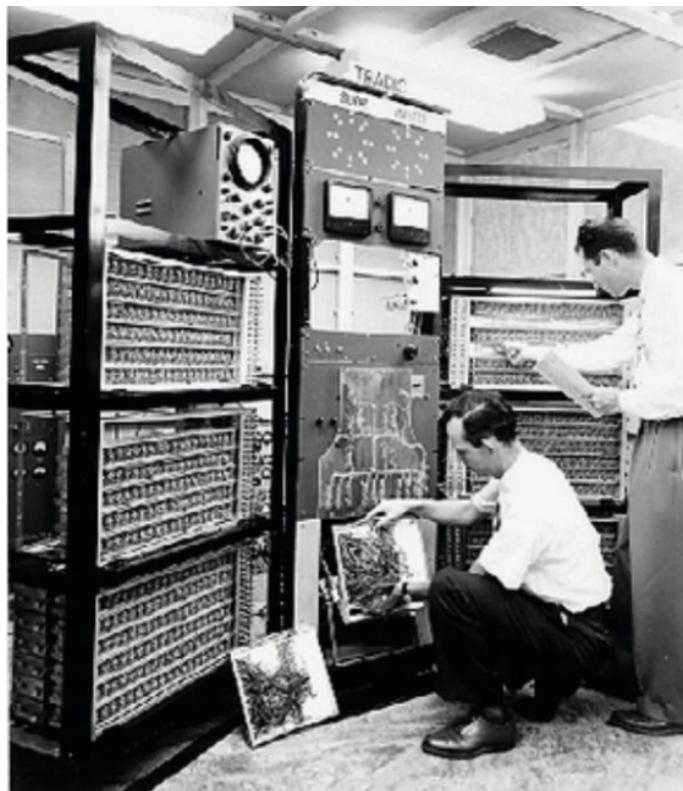


Bild 1.12 Elektronische Rechenmaschine mit Transistoren und Dioden, TRansistorized Airborne Digital Computer (TRADIC) (© Wikipedia, 2016) https://en.wikipedia.org/wiki/TRADIC#/media/File:TRADIC_computer.jpg

Der Aufbau einer Elektronikschaltung mit einzelnen Bauelementen und deren elektrische Verbindung mittels einer Leiterplatte wird als diskrete Elektronik bezeichnet. Durch die Entwicklung integrierter Schaltkreise (IC, Integrated Circuit) war es möglich, sehr viele elektronische Bauelemente auf einem Substratmaterial (Halbleiter) zu realisieren. Gleichzeitig konnten in diesem Material die elektrischen Verbindungen hergestellt werden. Der Platzbedarf für komplexere Schaltungen sank dadurch stark. Es entstanden die ersten Schaltkreise für Berechnungsaufgaben auf Basis dieser Technologie, sogenannte Mikroprozessoren. Der erste ab 1971 in Serie gefertigte und am freien Markt verfügbare Mikroprozessor (Bild 1.13) wurde von Intel® entwickelt. Er bestand aus 2250 Transistoren. Sie bilden bis heute den Kern einer Rechenmaschine bzw. eines Computers.

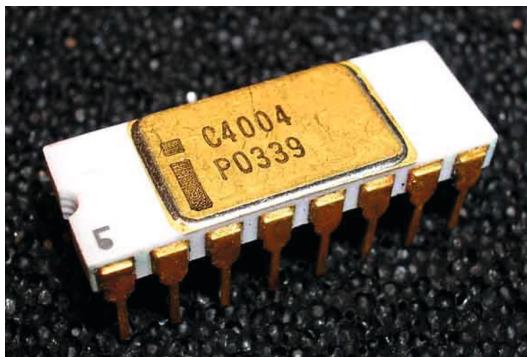


Bild 1.13 Erster in Serie gefertigter und am freien Markt verfügbarer 4-Bit-Mikroprozessor (Intel® 4004) (© Peter1912, Wikipedia, 2016) https://de.wikipedia.org/wiki/Intel_4004#/media/File:C4004_two_lines.jpg

Durch die Verfügbarkeit von Mikroprozessoren und deren rasante Weiterentwicklung konnten Computer sehr schnell stark verkleinert und in der Rechenleistung gesteigert werden. Gleichzeitig sanken die Kosten. Dies führte zur Marktreife von Computern für den Privatgebrauch und den Arbeitsplatz. Einen der ersten Heimcomputer, der von MITS in den Jahren 1974 bis 1975 in den USA entwickelt wurde (Altair 8800), zeigt Bild 1.14.



Bild 1.14 Einer der ersten auf einem Mikroprozessor basierenden Heimcomputer: Altair 8800 (© Michael Holley, Wikipedia, 2016) https://upload.wikimedia.org/wikipedia/commons/0/01/Altair_8800_Computer.jpg

Im Jahr 1965 hat Gordon Moore (einer der Gründer von Intel®) die Behauptung aufgestellt, dass sich die Anzahl an Transistoren pro Flächeneinheit, bei gleichen Kosten für die Halbleiterfläche, regelmäßig verdoppelt. Als Zeitraum für die Verdopplung wurden anfangs 12 Monate angegeben, mittlerweile wird von 18 bzw. 24 Monaten ausgegangen. Dieser Zusammenhang wird als Moore'sches Gesetz (Moore's Law) bezeichnet. Bis heute verhält sich die Entwicklung von Mikroprozessoren nach dieser Gesetzmäßigkeit. Das Moore'sche Gesetz ist kein Naturgesetz, sondern basiert auf der Beobachtung von Moore, dass die Kosten eines Halbleiterschaltkreises verfahrensbedingt mit sinkender und steigender Komponentenanzahl ansteigen. Bei niedriger Komponentenanzahl wird das Halbleitermaterial nicht voll ausgenutzt. Bei zu hoher Komponentenanzahl steigen die Fertigungskosten durch aufwändigeres Herstellungsverfahren bzw. niedrigere Ausbeuten (mehr Ausschuss) an. Dazwischen gibt es ein Kostenoptimum. Die von Moore aufgestellte Gesetzmäßigkeit bezieht sich immer auf das Kostenoptimum. Vereinfacht ausgedrückt steigt die Rechenleistung bei gleichen Kosten kontinuierlich an. Einige Experten gehen davon aus, dass die auf dem Moore'schen Gesetz basierende Entwicklung sich in einigen Jahren nicht mehr fortsetzen lässt.

■ 1.2 Computerunterstützung bei der Lösung mathematischer Aufgaben

Wie in vielen anderen Bereichen ist zur Lösung von Aufgaben bei der Produktentwicklung eine mathematische Beschreibung erforderlich. Ein sehr einfaches Beispiel für eine mathematische Beschreibung ist die horizontale Bewegung einer Masse, auf die in Bewegungsrichtung eine Kraft wirkt (Bild 1.15). Die Bewegungsgrößen, welche die Bewegung beschreiben, sind:

- Position
- Geschwindigkeit
- Beschleunigung

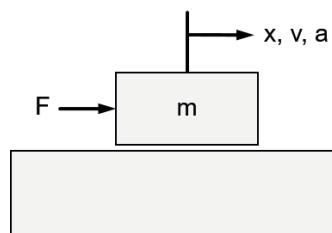


Bild 1.15 Bewegung einer Masse

Die mathematischen Zusammenhänge für die Bewegungsgrößen lauten ganz allgemein:

$$x(t) = \int v(t) dt \quad (1.1a)$$

$$v(t) = \int a(t) dt \quad (1.1b)$$

Bei konstanter Masse gilt für die Beschleunigung:

$$a(t) = \frac{1}{m} F(t) \quad (1.2)$$

x	Position	Position	m
v	Geschwindigkeit	Velocity	m/s
a	Beschleunigung	Acceleration	m/s ²
t	Zeit	Time	s
m	Masse	Mass	kg
F	Kraft	Force	N

Wirkt auf die Masse eine konstante Kraft in Bewegungsrichtung, so ist die Beschleunigung ebenfalls konstant und keine Funktion der Zeit:

$$a(t) = a = \frac{F}{m} \quad (1.3)$$

Damit kann die Lösung für den zeitlichen Verlauf der Position und der Geschwindigkeit direkt erfolgen (geschlossene Lösung). Unter der Annahme, dass die Bewegung zum Zeitpunkt $t = 0$ beginnt und die Position und Geschwindigkeit zu diesem Zeitpunkt jeweils null ist, gilt:

$$v(t) = \int a(t) dt = a \int_0^t dt = a t ; v(t=0) = 0 \quad (1.4a)$$

$$x(t) = \int v(t) dt = a \int_0^t t dt = \frac{1}{2} a t^2 ; x(t=0) = 0 \quad (1.4b)$$

Beginnt die Bewegung zum Zeitpunkt $t = t_1$ und hat die Position zu diesem Zeitpunkt den Wert x_1 und die Geschwindigkeit den Wert v_1 , gilt:

$$v(t) = a \int_{t_1}^t dt = a(t - t_1) + v_1 ; v(t=t_1) = v_1 \quad (1.5a)$$

$$x(t) = \int_{t_1}^t v(t) dt = \frac{1}{2} a(t - t_1)^2 + v_1(t - t_1) + x_1 ; x(t=t_1) = x_1 \quad (1.5b)$$

Ändert sich die Kraft beliebig mit der Zeit, so gibt es keine geschlossene Lösung zur Beschreibung der Bewegung. Für derartige mathematische Aufgaben werden Verfahren der numerischen Mathematik genutzt. Als Teilgebiet der Mathematik beschäftigt sich die numerische Mathematik mit der Entwicklung und der Analyse von Algorithmen für kontinuierliche mathematische Aufgaben. Numerische Verfahren liefern als Ergebnis Zahlenwerte.

Viele auf mathematischen Beschreibungen basierende Aufgaben von Ingenieuren lassen sich durch numerische Berechnungsverfahren lösen. Das Ergebnis einer numerischen