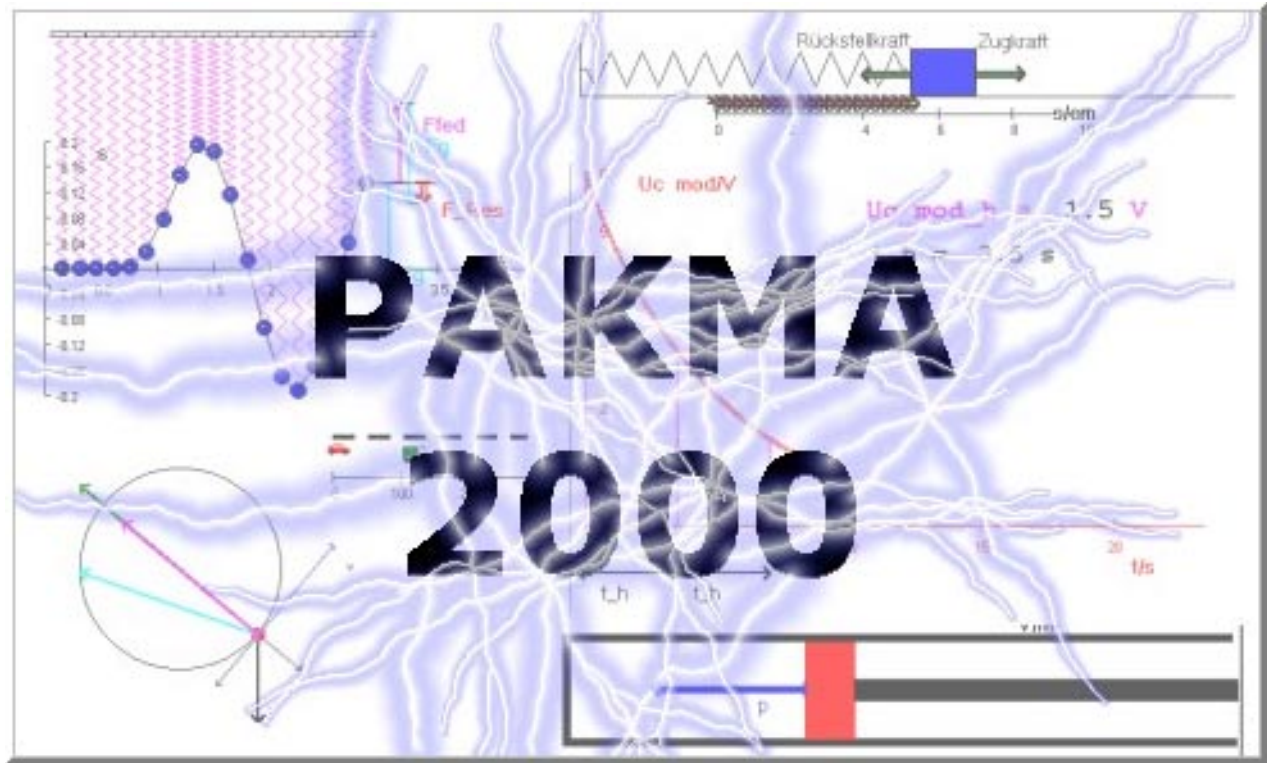


BENUTZERHANDBUCH



Teil I: Simulationen, Animationen erstellen Messungen durchführen

WÜRZBURG 2001

Neu überarbeitet von H. Dietrich

nach Vorversionen für PAKMA 2.x von H. Dietrich, E. Spiegel & P. Klein

Inhaltsverzeichnis

0 Einleitung.....	6
0.1 Vorwort: Drei Einsatzbereiche des PAKMA.....	6
0.2 Entwicklung von PAKMA und Dank.....	7
0.3 Zum Handbuch.....	9
1 Arbeiten mit fertigen Projekten	10
1.1 Grundlagen.....	10
1.1.1 Der Menüpunkt Datei.....	11
1.1.2 Der Menüpunkt Bearbeiten.....	11
1.1.3 Der Menüpunkt Eingaben.....	11
1.1.4 Der Menüpunkt Ausgaben.....	12
1.1.5 Der Menüpunkt Prog.-Lauf.....	13
1.1.6 Der Menüpunkt Fenster.....	13
1.2 Projekt öffnen.....	13
1.3 Projekte starten.....	14
1.4 Projekte verändern.....	16
1.4.1 Änderungen im Startwerte-Fenster.....	16
1.4.2 Änderungen am Kernprogramm vornehmen.....	17
1.4.3 Variablen selbst definieren.....	17
1.4.4 Ausgabe neu definierter Variablen.....	18
1.4.5 Bereichsgrenzen nach Änderungen anpassen.....	18
2 Erstellen von Projekten.....	20
2.1 Das Kernprogramm.....	20
2.1.1 Der Menüpunkt Bearbeiten.....	22
2.1.2 Der Menüpunkt Optionen	23
2.1.3 Der Aufbau eines Kernprogramms.....	24
2.1.4 Das Startwerte-Fenster.....	25
2.2 Die Bereichsgrenzen.....	26
2.2.1 Der Menüpunkt Bearbeiten.....	27
2.2.2 Für Sonderfälle: Optionen.....	27
2.3 PAKMA Anweisungen.....	29
2.4 Die Ausgabeelemente.....	30
2.4.1 Großanzeige.....	30
2.4.2 Zeigerinstrument.....	32
2.4.3 Balkendarstellung.....	33
2.4.4 Graphendarstellung.....	34
2.5 Schieber, Schalter und Taster.....	36

2.6 Die Animationselemente.....	37
2.6.1 Das Rechteck.....	38
2.6.2 Das Dreieck.....	40
2.6.3 Der Vektor.....	41
2.6.4 Breitpfeil.....	41
2.6.5 Feder.....	43
2.6.6 Bitmap.....	43
2.6.7 Benutzerdefinierte Bitmaps.....	43
2.6.8 Uhr.....	44
2.6.9 Text.....	44
2.6.10 Vielfachanimationselemente.....	44
2.6.10.1 Reihenvektor.....	44
2.6.10.2 Rechteckkette.....	47
2.6.11 Das Stempeln.....	49
2.6.12 Überblick über verwendete Elemente.....	50
2.6.13 Überblick über die eingetragenen Beziehungen.....	51
2.7 Erstellen des Projektes Federschwingung.....	52
2.7.1 physikalisches Modell erstellen	52
2.7.2 Kernprogramm schreiben.....	53
2.7.3 Startwerte in Menü eintragen.....	53
2.7.4 Bereichsgrenzen festlegen.....	54
2.7.5 Ausgaben-Fenster gestalten.....	55
2.7.5.1 Texte.....	57
2.7.5.2 Linie.....	57
2.7.5.3 Vektoren.....	58
2.7.6 Bedienungsfreundliche Ergänzungen im Kernprogramm.....	60

3 Der graphische Editor.....65

3.1 Vorgehensweise.....	65
3.2 PAKMA- Projekt laden.....	66
3.3 Wirkungsgefüge laden.....	66
3.4 Kernprogramm erzeugen.....	67
3.5 Programm sichern.....	68

4 Messen mit PAKMA.....70

4.1 Charakterisierung der Messbefehle von PAKMA.....	71
4.1.1 Messtypen.....	71
4.1.1.1 Zählen von Impulsen in einem Zeitintervall.....	71
4.1.1.2 Spannungen messen.....	71
4.1.2 Messschritte:.....	72
4.2 Das Ort-Zeit-Verhalten eines Federschwingers.....	72
4.3 Ladungsvorgänge am Kondensator.....	77

4.4 Magnetfeld längs der Achse von Kreisströmen.....	82
4.5 Computermouse als zweidimensionaler Bewegungssensor.....	85
4.6 Die Maus als eindimensionaler Low-Cost-Bewegungssensor.....	87
4.6.1 Versuchsbeispiel: Fahrbahnversuche.....	87
4.6.2 Versuchsbeispiel: Federpendel.....	88
4.7 Messen mit der seriellen Schnittstelle.....	89

5 PAKMA-Befehle und -Funktionen.....92

5.1 Variablen.....	92
5.2 Deklaration von Variablen.....	92
5.3 Schleifenbefehle und bedingte Ausführung von Anweisungen.....	92
5.4 Rechenoperationen.....	93
5.5 Logische Operationen.....	94
5.6 Sonstige Funktionen.....	94
5.7 Kernprogramm als procedure kern.....	94
5.8 Prozeduren und Funktionen.....	95
5.8.1 Beispiel.....	95
5.8.2 Abweichungen vom Pascal-Standard.....	95
5.9 PAKMA-spezifische Befehle.....	96
5.9.1 Anweisungen zur Initialisierung bzw. Rücksetzen.....	96
5.9.2 Messtypen.....	96
5.9.3 Einstellungen für alle Messtypen.....	96
5.9.4 Einstellungen für zählen.....	96
5.9.5 Einstellungen für mauszählen.....	97
5.9.6 Einstellungen für u_meter.....	97
5.9.7 Befehle zum Messen.....	97
5.9.8 Übergabe der Messwerte an Variablen.....	97
5.9.9 Einlesen von Eingabewerten während des Programmlaufes.....	97
5.9.10 Bereitstellen von Variablen für die Ausgabeelemente.....	98
5.9.11 Warte-Routine.....	98
5.9.12 Befehle zum Steuern über die Interfacebox.....	98
5.9.13 Funktionen zum Abfragen des Programm-Modus.....	98
5.9.14 Zeitverzögerung.....	98
5.9.15 Graphensteuerung.....	98
5.9.16 Neue externe Funktionen und Prozeduren in der Unit Pak_math.....	98

6 Anhang.....100

6.1 Fehlermeldungen.....	100
6.2 Installations- und Laufprobleme.....	101
6.2.1 Einstellungen vornehmen.....	101
6.2.2 Bekannte Laufprobleme.....	102

6.3 Bezugsquellen.....	102
6.4 Rückmeldungen / „Hotline“	102
6.5 PAKMA im Internet.....	102
6.6 Literatur zu PAKMA	104

0 Einleitung

0.1 Vorwort: Drei Einsatzbereiche des PAKMA

Am Lehrstuhl für Didaktik der Physik der Universität Würzburg wurde über mehrere Jahre das Softwaresystem PAKMA (Physik Aktiv Messen, Modellieren, Analyse, Animation) entwickelt und erweitert. Das Ziel war dabei, dem Benutzer die Möglichkeit zu geben, seine Ideen für die Rechnerunterstützung mit einem möglichst universell einsetzbaren Werkzeug zu realisieren.

PAKMA ist als ein offenes Softwaresystem geschaffen worden. Ein Schwerpunkt ist dabei, Versuchsabläufe zu den verschiedensten physikalischen Fragestellungen mit dem Computer aufnehmen und aufbereiten zu können. Hier stehen dem Benutzer verschiedene Möglichkeiten zur Verfügung, die mit PAKMA erfaßten Messwerte zeitgleich zum Experiment möglichst aussagekräftig darzustellen. Zum einen können relevante Versuchsaussagen, wie z.B. funktionale Abhängigkeiten, in Diagrammen oder anderen Anzeigeformen veranschaulicht werden. Weiterhin besteht die Möglichkeit, den Vorgang des physikalischen Experiments auf dem Bildschirm durch ikonisch dynamische Repräsentation zu visualisieren. So kann beispielsweise die Schwingung einer Masse an einer Feder parallel zum realen Experiment auf dem Bildschirm durch das Anzeigen von bewegten Animationselementen dargestellt werden, die dann z.B. auch Geschwindigkeit, Beschleunigung und Kräfte als Vektoren anzeigen. In der Reproduktion können dann mit aufgenommenen Versuchsdaten Versuchsabläufe auf dem Bildschirm wiedergegeben, auch in Zeitlupe betrachtet und in einer Abfolge von Standbildern detailliert analysiert werden.

Der bisher am häufigsten genutzte Einsatzbereich des PAKMAs ist das Zeigen und Analysieren von fertig erstellten Simulationen. Hier werden physikalische Phänomene und ihre Abhängigkeit von physikalischen Größen am Bildschirm schematisch dargestellt. Wie bei der Aufbereitung experimenteller Daten werden physikalische Größen und relevante Zusammenhänge durch dynamisch ikonische Repräsentationen visualisiert. Das PAKMA-System kann aber nicht nur dazu eingesetzt werden, um Messwerte und Experimente zu fassen und diese dann zu analysieren, an ihnen physikalische Phänomene zu studieren und in ihren Abhängigkeiten kennen zu lernen.

In einem dritten Einsatzbereich können Lernende mit PAKMA auch physikalische Modelle erstellen. Hierbei kann man beliebige physikalische Sachverhalte numerisch iterativ berechnen lassen und diese ebenfalls durch graphische Darstellungen nämlich durch Diagramme, aber auch dynamisch ikonische Repräsentation darstellen. Diese Modellbildung ist nicht nur gleichungsorientiert möglich. Seit kurzem besteht nämlich mit dem Zusatzmodul VisEdit auch die Möglichkeit, Wirkungsgefüge am Bildschirm zu entwerfen und dies automatisch in ein lauffähiges Kernprogramm umzusetzen. Gegenüber anderen Modellbildungssystemen bietet PAKMA mit dieser Erweiterung zwei entscheidende Vorteile: Einerseits können auch Messvorgänge und ihre Auswertung mit in zu erstellende Netzwerkgefüge integriert werden, so daß Messwerterfassung mit Auswertung und Modellbildung parallel ablaufen und ihre Ergebnisse in Realzeit verglichen werden können, so dass sich sofort Aussagen über die Angemessenheit des zugrunde gelegten Modells ergeben. Andererseits können die Ergebnisse eines ablaufenden Modells nicht nur durch Graphen sondern auch durch dynamisch ikonische Repräsentationen visualisiert werden.

Mehr Details zum PAKMA und seinen Einsatzmöglichkeiten finden Sie im PAKMA_Infoheft.

0.2 Entwicklung von PAKMA und Dank

D. Heuer

Die PAKMA-Konzeption wurde zuerst am C64 mit der Sprache COMAL 2.0 realisiert. Hier hat Herr Roland Treffer, Würzburg, intensiv an der Umsetzung der Vorstellungen gearbeitet, während Herr Heino Sand, Erlangen die Messroutinen erstellte. Parallel dazu wurden neue Versuchsideen verwirklicht, die in Zeitschriften erschienen, s. z. B. in der Serie: Effizientes Experimentieren und Analysieren mit PAKMA, in PdN-Ph ab Heft 7, 1989. Mit erweiterten Gestaltungsmöglichkeiten wurde PAKMA dann auf den Amiga übertragen, als er noch der preiswerte Grafikrechner war. Herr M. Vaeth und Herr H. Graefe verwirklichten das Grundkonzept, Herr H. Sand wiederum die Messroutinen. Neue Animationsmöglichkeiten schuf Herr M. Schröder. Versuche mit solchen ikonischen Repräsentationen wurden zuerst 1992 vorgestellt.

Im Rahmen eines Studienvertrages mit großzügiger finanzieller Unterstützung der Firma IBM Deutschland war es dann möglich, eine PAKMA-Implementation am PC vorzunehmen, anfänglich unter DOS. Bald wurden die Arbeiten aber unter WINDOWS 3.1 weitergeführt. Die Grundkonzeption realisierte Herr St. Hild, einen Teil der Messroutinen Herr Th. Hahn. Die grafisch editierbaren Animationsmöglichkeiten implementierte Herr M. Schmidt. Versuchsvorschläge mit solchen ikonischen Repräsentationen wurden mit PAKMA ab 1994 in Zeitschriftenaufsätzen veröffentlicht. Weitere Ausgestaltungen des PAKMA für Windows nahmen Herr M. Prautzsch, Herr P. Klein und Herr V. Phan Ghia vor. Herr Dipl. Math. Ch. Kahmann hat seit 1995 viele Verbesserungen und wichtige Erweiterungen realisiert. Seit 1998 hat Herr Sebastian Bärhausen das PAKMA engagiert weiterentwickelt und auch die neue Version PAKMA 2000 erstellt, die neben vielen neuen Bedienungs- und Gestaltungselement als 32-bit Version auch unter Windows NT lauffähig ist.

Der langen Reihe von Mitarbeitern sei auch an dieser Stelle für ihr engagiertes Arbeiten herzlich gedankt, z. B. denen die an diesem Handbuch mitgearbeitet haben, auch solchen, die hier nicht namentlich erwähnt sind, insb. denen, die in Zulassungs- oder Hausarbeiten mit PAKMA gearbeitet haben. Ohne sie alle hätten die bestehenden Vorstellungen nicht umgesetzt werden können.

Ebenso bin ich den Kollegen aus unserem Würzburger Physik-Computerarbeitskreis und weiteren interessierten Kollegen zu Dank verpflichtet, die mit wohlwollender Anerkennung und vielfachen Anregungen aus ihren Unterrichtseinsätzen das PAKMA-Projekt gefördert haben.

0.3 Systemvoraussetzungen und Installation

PAKMA läuft unter der Benutzeroberfläche Windows und wird demzufolge im wesentlichen mit der Maus, aber auch mit der Tastatur des Computers bedient. Um mit PAKMA arbeiten zu können, benötigt man keine speziellen Computerkenntnisse. Gewisse Fertigkeiten bei der Handhabung von Fenstern und Menüleisten sind jedoch hilfreich.

PAKMA 2000 ist ein 32-Bit Programm und läuft ausschließlich unter Windows 95 oder höheren Versionen. Es sollten mindestens 16 MB Arbeitsspeicher und ein Pentium Prozessor vorhanden sein.

Sollen mit dem PAKMA- System Messungen über die PAKMA-Interface Box durchgeführt werden, wird außerdem Platz für den Einbau einer ISA-Messkarte benötigt. Die Bezugsadresse für einzusetzende Hardware ist im Anhang dieses Handbuches angegeben.

Die Installation der PAKMA- Software erfolgt vollständig menügesteuert. Ist die PAKMA- Software auf einer CD vorhanden, so erscheint nach deren Einlegen ein Menü, in dem der Punkt **PAKMA installieren** ausgewählt werden kann.

Das Setup- Programm kann ebenfalls gestartet werden, indem die Datei **setup.exe** im PAKMA- Verzeichnis doppelt angeklickt wird. Eine weitere Möglichkeit ist das direkte **Ausführen** der Datei **setup.exe** mit Hilfe der entsprechenden Windows- Funktion.

Nach dem Start erscheint ein Fenster, in dem das gewünschte Installationsverzeichnis bestimmt werden kann (s. **Abb. 0.1**). Wird der Button **Durchsuchen...** gedrückt, kann eine anderes Verzeichnis als das vorgegebene gewählt werden.

Nach Drücken der Taste **Weiter** wird die PAKMA- Software automatisch auf dem Rechner installiert.

Nach erfolgreicher Installation erscheint eine entsprechende Meldung auf dem Bildschirm.

*Da im Moment leider noch **keine Deinstallationsroutine** implementiert ist, werden bestehende PAKMA-Versionen dadurch entfernt, dass das **PAKMA-Verzeichnis** gelöscht wird.*

Sollen vorhandene Projekte weiterhin verwendet werden, ist es ratsam, diese in ein separates Verzeichnis zu kopieren bevor man mit dem Installieren beginnt.

Möchte man Projekte verwenden, die mit älteren Versionen der PAKMA- Software erstellt wurden, so erscheint beim Öffnen dieser Projekte ein entsprechender Hinweis. Speichert man diese Projekte mit der PAKMA- Version 2000 erneut ab, entfällt zukünftig diese Meldung. Es können jedoch einige optische Veränderungen im Ausgaben-Fenster des alten Projektes aufgetreten sein, die aber an der prinzipiellen Funktionsweise dieses Projektes nichts ändern sollten.



Abb. 0.1



0.3 Zum Handbuch

Um einen Überblick über den Aufbau des vorliegenden Handbuchs zu bekommen, werden im Folgenden die einzelnen Kapitel kurz vorgestellt. Der Leser kann so bei Bedarf einzelne Kapitel gezielt auswählen.

Kapitel 1

Dieses Kapitel vermittelt die grundlegenden Eigenschaften von PAKMA. Es wird gezeigt, wie man mit bereits existierenden Projekten arbeitet und dort kleine Veränderungen vornehmen kann.

**Mit fertigem
Projekt arbeiten**

Kapitel 2

Hier wird ausführlich beschrieben, wie man selbst Projekte für PAKMA erstellt. Anhand eines Beispielprojektes wird dies Schritt für Schritt vorgeführt.

Projekt erstellen

Kapitel 3

Dieses Kapitel soll vermitteln, wie das PAKMA-System zum Messen von physikalischen Größen eingesetzt werden kann. Auch Messungen mit Hilfe der Computermouse und über die serielle Schnittstelle werden erläutert.

**Messwerte
aufnehmen**

Kapitel 4

Dieses Kapitel listet die Befehle und Funktionen von PAKMA auf.

**Befehle und
Funktionen**

Anhang

Der Anhang enthält wichtige Informationen zu den Themen Fehlermeldungen, Laufprobleme, Bezugsquellen, Rückmeldungen und PAKMA im Internet.

**Fehler-meldun-
gen**

Da ein „learning by doing“ sehr effizient ist, wenn es darum geht, Handlungsabläufe kennenzulernen, ist es empfehlenswert, die beschriebenen Sachverhalte parallel zum Lesen praktisch am Computer zu erproben. Es empfiehlt sich daher, dieses Handbuch in gedruckter Form zu verwenden.



1 Arbeiten mit fertigen Projekten

Für den Einsatz von PAKMA sind bereits viele komplette Projekte zu den verschiedensten Teilgebieten der Physik vorhanden. Man muss also keine eigenen Projekte erstellen, um mit PAKMA arbeiten zu können.

Im Literaturverzeichnis dieses Handbuches werden die bisher erschienenen CDs, die PAKMA- Projekte enthalten, aufgeführt. Diese stellen neben dem Internet Möglichkeiten dar, fertige Projekte zu erhalten.



In diesem Kapitel soll nun der Umgang mit bereits erstellten Projekten erläutert werden. Außerdem wird gezeigt, wie kleinere Änderungen an diesen bestehenden PAKMA-Projekten vorgenommen werden können.

1.1 Grundlagen

Zentraler Bestandteil von PAKMA 2000 sind die in **Abb. 1.1** dargestellte Symbol- und die Menüleiste. Zwar lassen sich sämtliche Funktionen von PAKMA als Unterpunkte der jeweiligen Menüpunkte aufrufen, die Bedienung mit Hilfe der Icons ist aber wesentlich komfortabler.

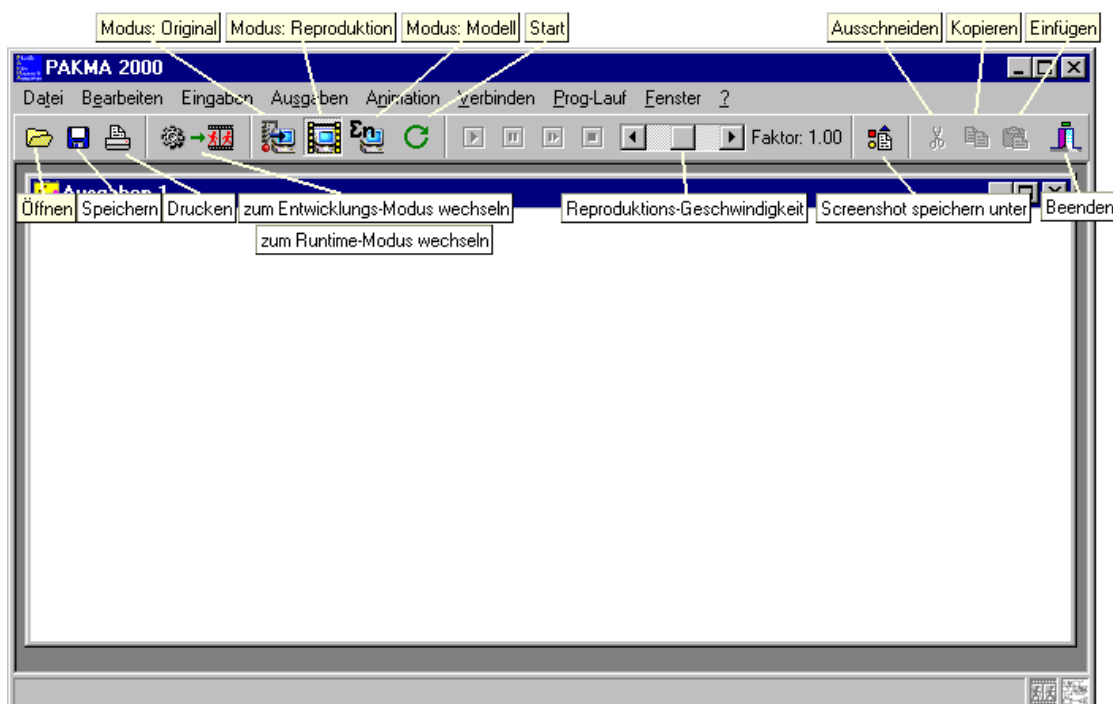


Abb. 1.1

Obwohl die Symbole die zugehörige Funktion veranschaulichen sollen, erhält man einen entsprechenden Hinweis eingeblendet, wenn sich der Mauszeiger über dem jeweiligen Icon befindet.



Detaillierte Informationen über die Handhabung der einzelnen Icons werden in den nun folgenden Abschnitten mitgeteilt. Doch zunächst sollen die einzelnen Menüpunkte kurz vorgestellt werden.

1.1.1 Der Menüpunkt Datei

Über den Menüpunkt **Datei** gelangt man zu den Funktionen, die die Verwaltung von Projekten bzw. einzelner Module ermöglichen. **Abb. 1.1** zeigt, welche Unterpunkte nach Anklicken des Menüpunktes **Datei** zur Auswahl stehen. Die Punkte **Neu**, **Öffnen**, **Speichern**, **Speichern unter** und **Beenden** haben die unter Windows üblichen Funktionen und beziehen sich auf komplette PAKMA-Projekte.

Die Unterpunkte **Modul laden** bzw. **Modul speichern** dienen dazu, die zuvor schon angesprochenen Module einzeln zu laden bzw. zu speichern.

Der Unterpunkt **Messdaten exportieren** ermöglicht das Abspeichern von Messwerten im ASCII-Format. Damit lassen sich aufgenommene Messwerte, unabhängig von PAKMA, z. B. in ein Tabellenkalkulationsprogramm exportieren, um sie dort auswerten, bearbeiten bzw. darstellen zu können.

Projekt-Info starten ... bzw. **Videoclip starten ...** sind Funktionen, die eine Rolle spielen, wenn man PAKMA ab Version 2.5 zusammen mit einer CD erhalten hat, auf der sich zugehörige **Projektbeschreibungen** im HTML-Format bzw. **Videoclips** in Form von *.avi oder *.mov Dateien befinden. Nach dem Anklicken startet der Dialog **Datei öffnen**, der ein Laden dieser Daten ermöglicht.

Mit dem Befehl Speichern werden – wie unter Windows üblich – gleichnamige Projekte, die bereits existieren, ohne Vorwarnung überschrieben!

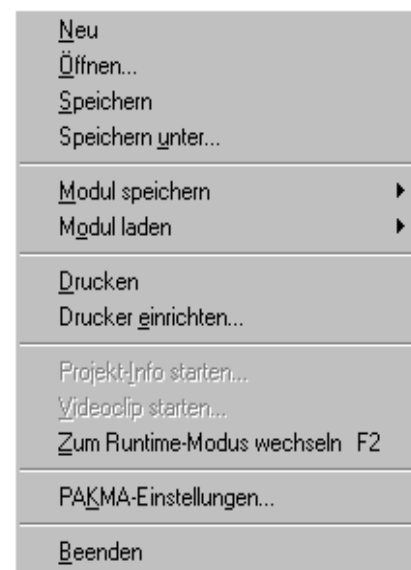


Abb. 1.1


Achtung

1.1.2 Der Menüpunkt Bearbeiten

Mit diesem Menüpunkt werden die in Windows üblichen Funktionen **Ausschneiden**, **Kopieren** und **Einfügen** zur Verfügung gestellt.

Neu Zeichnen dient dazu, den Inhalt der Ausgaben-Fenster zu aktualisieren, eine Funktion, die vor allem dann benötigt wird, wenn dort Veränderungen vorgenommen werden.

Mit dem Befehl **Externes Kernprogramm übernehmen** bietet PAKMA die Möglichkeit, Kernprogramme, die mit anderen Editoren erstellt und in die Zwischenablage kopiert wurden, in den Programmeditor von PAKMA zu übertragen.

Der Unterpunkt **VisEdit laden** kommt dann zum Einsatz, wenn mit dem **graphischen Editor** gearbeitet werden soll.

*Dieses Zusatzprogramm ist seit 1999 erhältlich. Ein Beispiel zur Einsatzweise dieses Editors wird in **Kapitel 3** kurz vorgestellt. Eine ausführliche Anleitung ist dem entsprechenden Handbuch zu entnehmen.*

Ausschneiden	Umsch+Entf
Kopieren	Strg+Einf
Einfügen	Umsch+Einf
Neu zeichnen	F3
VisEdit laden	
Externes Kernprogramm übernehmen	

Abb. 1.2


Hinweis

1.1.3 Der Menüpunkt Eingaben

Dieser Menüpunkt ist schon in diesem Kapitel von großer Bedeutung. Mit seiner Hilfe gelangt man zu den Punkten **Programmeditor**, **Startwerte** und **Bereichsgrenzen**, über die Projekte direkt veränderbar sind.

Schieber/Schalter stellen eine Möglichkeit dar, den Wert von Variablen während eines Programmlaufs direkt in einem Ausgaben-Fenster zu verändern. In diesem Kapitel wird beschrieben, wie solche Schieber und Schalter in fertigen Projekten zum Einsatz kommen.

Programmeditor	F5
Startwerte	F6
Bereichsgrenzen	F7
Schieber/Schalter/Taster	
Variablen definieren	
Serielle Geräte definieren...	

Abb. 1.3

Über den Punkt **Variablen definieren** hat man die Möglichkeit, neue Variablen in ein Projekt aufzunehmen. Diese müssen sich jedoch aus bereits existierenden Ausgabegrößen berechnen lassen.

1.1.4 Der Menüpunkt Ausgaben

Nachdem in einem Projekt die **Eingaben** festgelegt sind, kann man über die Unterpunkte des Menüpunktes **Ausgaben** die Ausgabegrößen eines oder mehrerer Ausgaben-Fenster auf unterschiedliche Weise visualisieren.

Wird der Unterpunkt **Neues Ausgabenfenster** aufgerufen, so wird ein neues Ausgaben-Fenster geöffnet. Innerhalb eines Projektes kann man mit verschiedenen Ausgaben-Fenstern arbeiten, welche zur Unterscheidung in der Statusleiste fortlaufend durchnummeriert werden. Alternativ kann man jedem Ausgaben-Fenster einen erklärenden Namen geben. Dazu ist ein entsprechender Dialog durch Drücken der rechten Maustaste bzw. durch Anklicken des Unterpunktes **Ausgabenfenster-Optionen** aufzurufen, s. Abb. 1.4.

Man kann hier neben dem **Titel** des Ausgaben-Fensters auch die genaue **Größe** des Fensters festlegen und ein **Hintergrundbild** zuweisen, s.

Abb. 1.5. Das Hintergrundbild muss hierbei als **BMP-Datei** vorliegen und darf nicht mehr als 256 Farben besitzen.

Durch Drücken der Taste **Zuweisen** gelangt man zu einem Dialog, mit dessen Hilfe das gewünschte Bild aus dem entsprechenden Verzeichnis geladen werden kann. Die genaue Position des Hintergrundbildes wird durch die Wahl des **x-Ursprungs** und des **y-Ursprungs** angegeben. Dies legt die Position der linken oberen Ecke des Bildes im Ausgaben-Fenster fest.

Mit Hilfe der Taste **Löschen** kann das Hintergrundbild eines Ausgaben-Fensters wieder entfernt werden. Soll ein bestehendes Hintergrundbild als Bitmap exportiert werden, so steht nach Drücken der **Exportieren**-Taste ein entsprechender Dialog **Speichern Unter** zur Verfügung.

Wird PAKMA neu gestartet, ist die Option **Bildzwischenspeicher** aktiviert, d. h. sämtliche Ausgaben, die in einem Ausgaben-Fenster dargestellt werden, werden zunächst in einen Zwischenspeicher abgelegt und anschließend komplett angezeigt. Dies bewirkt, dass die Bewegungen am Bildschirm flüssiger erscheinen. Hierzu ist allerdings ein größerer Rechenaufwand nötig. Man kann daher bei zeitkritischen Projekten bzw. langsameren Rechnern diese Funktion deaktivieren. Die gewählte Einstellung bleibt nach dem Abspeichern des jeweiligen Projektes erhalten.

Lässt man das gleiche Projekt auf unterschiedlich schnellen Rechnern ablaufen, so sollte man auf diese Einstellung besonders achten.

Ist der Unterpunkt **Graphendaten immer zeichnen** aktiviert, so werden vorhandene Graphendaten auch dann dargestellt, wenn Änderungen an Projekten im Editiermodus vorgenommen werden. Dies ermöglicht unter Umständen ein komfortableres Arbeiten.

Die Funktion **Screenshot speichern unter...** ermöglicht das Abspeichern der aktuellen Darstellung im aktiven Ausgaben-Fenster mit Hilfe eines entsprechenden Dialoges.

Damit fertiggestellte Projekte nicht versehentlich verändert werden, ist die Option **Editieren** in PAKMA implementiert. Man kann Ausgaben-Fenster nur dann verändern, wenn dieser

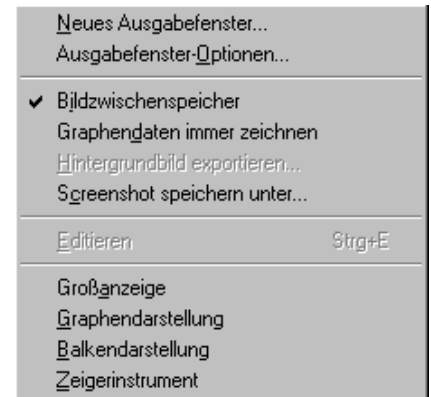


Abb. 1.4



Abb. 1.5



Hinweis

Punkt aktiviert ist. Man erkennt dies daran, dass die Funktion **Editieren** im **aktivierten** Zustand sonst nur **schemenhaft** sichtbar ist.

Die Unterpunkte **Großanzeige**, **Graphendarstellung**, **Balkendarstellung** und **Zeigerinstrument** ermöglichen das Visualisieren von Ausgabegrößen auf unterschiedliche Art und Weise.

*Die Menüpunkte **Animation** und **Verbinden** spielen erst beim Erstellen von dynamischen Repräsentationen eine Rolle und werden daher erst in **Kapitel 2** behandelt.*



1.1.5 Der Menüpunkt Prog.-Lauf

Die Unterpunkte des Menüpunktes **Prog-Lauf** dienen zum Starten der Projekte. Wie man dabei im Einzelnen vorzugehen hat, wird in **Abschnitt 1.3** geschildert.

Start als Original	F9
Start als Reproduktion	F10
Start als Modell	F11
Erneut starten	F12
Dialog für Graphik-Datenablage	

Abb. 1.6

1.1.6 Der Menüpunkt Fenster



Das Untermenü des Menüpunktes **Fenster** bietet verschiedene Möglichkeiten wie mehrere geöffnete Fenster dargestellt werden können. Die Unterpunkte **Überlappend** und **Nebeneinander** haben dabei die in Windows üblichen Bedeutungen.

Nebeneinander
Überlappend
Ausgabefenster-Reihenfolge ändern
Ausgabefenster-Reihenfolge wiederherstellen
✓ 1 Ausgaben 1

Abb. 1.7

1.2 Projekt öffnen

Prinzipiell kann man ein Projekt dadurch aufrufen, dass man im Windows- Explorers Dateien mit der Endung ***.prj** doppelklickt. Um welches Projekt es sich dabei handelt, sollte am Namens Kürzel zu erkennen sein.

Wurde PAKMA durch Anklicken des Symbols  bereits gestartet, kann man Projekte nach dem Klicken von  oder mit Hilfe des Unterpunktes **Öffnen** des Menüpunktes **Datei** aufrufen. In beiden Fällen erscheint folgender Dialog (s. Abb. 1.8). **Abb. 1.9** zeigt das aufgerufene Projekt **Feder-schwingung.prj**. Man kann unter dem Menüpunkt **Datei** mit Hilfe des Unterpunktes **Projekt-Info starten** die zugehörige Projektbeschreibung auswählen und öffnen.

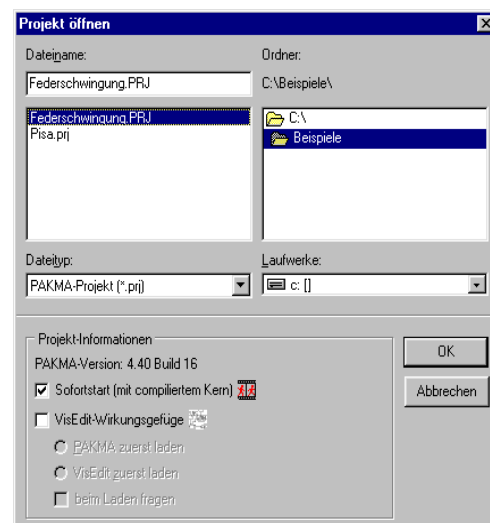


Abb. 1.8

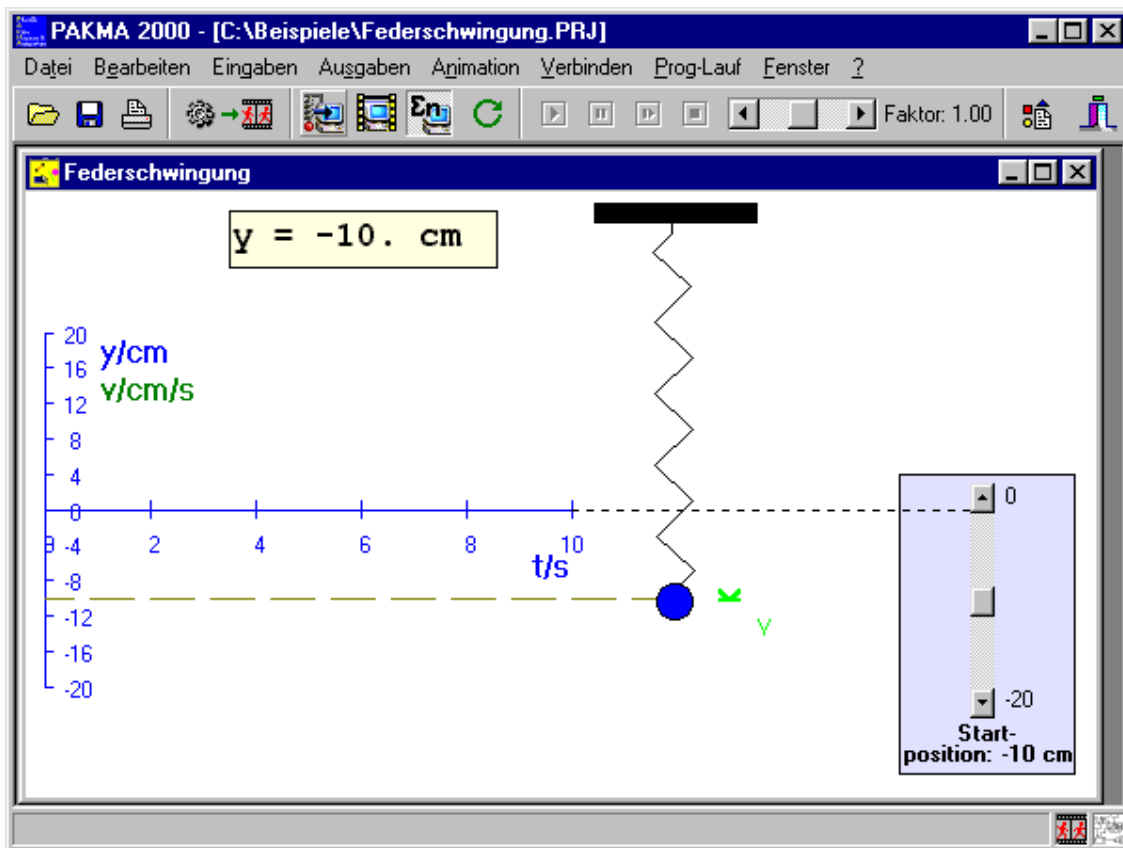


Abb. 1.9

1.3 Projekte starten

Nach dem Öffnen eines Projektes befindet sich dieses meistens schon im **Ablaufmodus**, d.h. es kann sofort mit Hilfe des in **Abb. 1.10** gezeigten Icons **Ablauf** gestartet werden. Zuvor kann man jedoch mit Hilfe evtl. vorhandener Schieber Einstellungen vornehmen, wobei die so entstehende Startsituation entsprechend visualisiert wird.

Läuft ein Projekt bereits, dienen die restlichen Icons in der Symbolleiste dazu, den jeweiligen Ablauf zu steuern.

Pause hält den Programmlauf an.

Einzelschritt führt das Programm schrittweise aus.

Stopp bricht den Programmlauf vorzeitig ab.

Ablauf lässt ein angehaltenes Projekt wieder weiterlaufen.

Es kann vorkommen, dass ein Projekt so abgespeichert wurde, dass es sich nicht automatisch im Ablaufmodus befindet und sich demnach nicht mit Ablauf starten lässt. Hier muss zunächst der gewünschte Ablaufmodus ausgewählt und dann das Projekt gestartet werden.



Abb. 1.10



Mit Hilfe der in **Abb. 1.11** dargestellten Icons kann der gewünschte Ablaufmodus gewählt werden. Durch Drücken der **Start-Taste** mit dem Icon wird dann das Kernprogramm kompiliert und das Projekt gestartet. Nun kann, wie zuvor beschrieben, der physikalische Ablauf am Bildschirm mit **Ablauf** gestartet werden.

Je nachdem welche Art von Projekt vorliegt, wählt man eine der folgenden Möglichkeiten aus.

- ⑩ Ablaufmodus als **Original** kommt nur dann zum Einsatz, wenn parallel zum Ablauf des Projektes Messdaten eines gleichzeitig ablaufenden Real-experimentes aufgenommen und aufbereitet werden sollen.


Ist ein aktives Messen nicht möglich oder erwünscht, ist eine der beiden anderen Startformen zu wählen.

- ⑩ Der Modus **Reproduktion** führt das Kernprogramm beim Programmstart ebenfalls neu aus, ermöglicht aber zusätzlich, die Ablaufgeschwindigkeit zu regulieren. Man kann dies mit dem Schieber **Faktor**, der in Abb. 1.12 dargestellt ist, in der Symbolleiste auch während des Laufs einstellen. Diese Startform kommt bei „**Experimental**“-Projekten zum Einsatz, wenn das Projekt mit bereits existierende Messdaten ablaufen soll.

Natürlich kann ein Projekt nicht schneller ablaufen, als es die Rechnerkapazität zulässt.

- ⑩ Bei dem Modus **Modell** läuft das Projekt, wenn es gestartet wird, mit der höchstmöglichen Geschwindigkeit des verwendeten Computers ab. Wie der Name schon sagt, wird diese Startform für **Modelle**, d. h. für Projekte ohne Messungen, verwendet.

*Wird ein **Modell** im Reproduktion-Modus gestartet, wird zwischen zwei Ausgabeaufforderungen ca. 0,1 s gewartet, wenn am **Repro- Schieber** keine Veränderung vorgenommen wird. Diese Zeit kann durch den eingestellten Faktor entsprechend verändert werden.*

Oft ist es vorgesehen, dass man in einem Projekt mehrere Durchläufe nacheinander ausführt. Dazu muss nach durchgeführtem Ablauf die Taste  erneut betätigt werden. Zuvor sollten jedoch, falls vorhanden, Veränderungen mit Hilfe der entsprechenden Schieber vorgenommen werden, ansonsten in den Startwerten. Es lassen sich so auf komfortable Art und Weise verschiedene Durchläufe auf dem selben Bildschirm dokumentieren und miteinander vergleichen.

Natürlich kann der Start eines Projektes auch mit Hilfe der Unterpunkte des Menüpunktes **Prog-Lauf** geschehen.

*Für PAKMA- Projekte, die die einschließlich Version 2.7 ohne Ablauf-Taste konzipiert wurden, war eine simplere Möglichkeit vorgesehen, die Graphen von verschiedenen Durchläufen auf dem selben Bildschirm zu vergleichen. Ist der Unterpunkt **Dialog für Graphenablage** des Menüpunktes **Prog-Lauf** aktiviert, erscheint bei jedem erneuten Start das Fenster, das in Abb. 1.13 zu sehen ist. Damit können die Daten der Graphen des letzten Durchlaufs in dafür vorgesehene Ablagen gespeichert werden. Zusätzlich zur Ablage der Graphendaten, kann man entscheiden, ob diejenigen Graphen, die sich bereits in einer Ablage befinden, zusätzlich zu den neu erstellten Graphen ausgegeben werden sollen.*



Ablaufmodus:

als **Original**

als **Reproduktion**

als **Modell**

Projekt starten

Abb. 1.11



Faktor: 1.00

Abb. 1.12



Hinweis



Hinweis

Hierbei ist jedoch zu beachten, dass sich die ursprünglichen Darstellungsmerkmale (z. B. Farbe und Form) nicht weiter verändern lassen.

Das Einbeziehen bestehender Graphen ist jedoch nur dann sinnvoll, wenn Änderungen im Projekt vorgenommen wurden, indem z. B. Schieber verändert wurden.

1.4 Projekte verändern

Schieber zu benutzen ist die einfachste Möglichkeit, veränderte Abläufe mit einem Projekt zu erreichen. Darüber hinaus kann man auch durch kleinere Eingriffe Projekte den eigenen Bedürfnissen anpassen, ohne gleich ein neues Projekt erstellen zu müssen.

Erreicht wird dies mit dem Menüpunkt **Eingaben**, der mit seinen Unterpunkten **Programmeditor**, **Startwerte**, **Bereichsgrenzen** bzw. **Variablen definieren** entsprechende Funktionen bereitstellt. Anhand des Projekts **Federschwingung.prj** soll dies im Folgenden gezeigt werden.

Prinzipiell berechnet jedes Projekt die auszugebenden Größen iterativ in seinem Kernprogramm. Die dazu benötigten Anfangswerte und Konstanten können in PAKMA entweder in einem speziell dafür vorgesehenen Startwerte-Fenster oder direkt im Kernprogramm definiert werden.

1.4.1 Änderungen im Startwerte-Fenster

Startwerte, die der Benutzer evtl. ändert, um mit dem Projekt explorativ umzugehen, sollten als entsprechende Zuweisungen in der Startwertetabelle aufgeführt sein. Hier sind die Anfangswerte und Konstanten übersichtlich zu verwalten. Indem man aus dem Menüpunkt **Eingaben** den Unterpunkt **Startwerte** aufruft bzw. die Taste **F6** drückt wird das Startwerte-Fenster geöffnet (s Abb. 1.14).

Der in dieser Spalte eingetragene Wert wird vom Programm für den nächsten Ablauf verwendet. Dieser Wert kann beliebig verändert werden.

Möchte man nun den Zahlenwert für eine Variable verändern, so klickt man das Feld in der Spalte **Neuer W.** für die entsprechende Variable an und gibt den gewünschten Wert ein. Neben Zahlenwerten können hier auch einfache mathematische Operationen, wie z. B. $\frac{2}{3}$ bzw. $2 \cdot 3$ eingetragen werden.

Die Eingabe in die Spalte **Neuer W.** wird durch die Schaltflächen **Vorgabe** bzw. **Letzter Wert** vereinfacht. Klickt man eine dieser Schaltflächen an, wird der gesamte Inhalt der entsprechenden Spalte in die Spalte **Neuer W.** übertragen. Dann sind nur noch die Werte zu verändern, die nicht übernommen werden sollen. Ist in der Spalte **Neuer W.** kein Zahlenwert für eine Größe vorhanden, wird für den folgenden Durchlauf der Wert der Vorgabe verwendet. Sind sämtliche Änderungen vorgenommen, kann das Fenster **Startwerte** geschlossen werden, die Werte werden automatisch übernommen.

Normalerweise sollte sich ein Projekt nach dem Start im **Sofortstartmodus** befinden. Dies wird dadurch erreicht, dass beim Abspeichern eines Projektes die entsprechende Funktion aktiviert wird. Es wird im Folgenden immer wieder auf

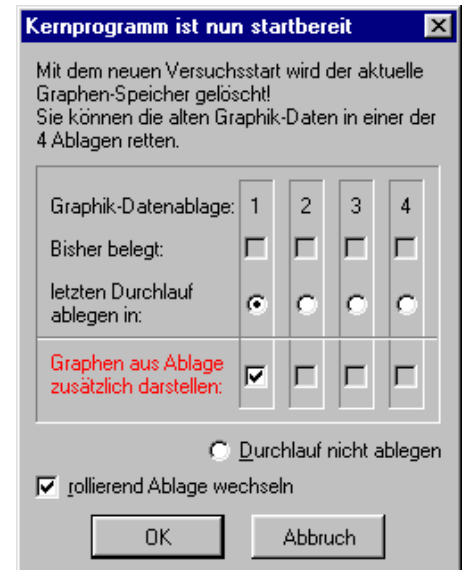


Abb. 1.13

Startwerte (nur Zahlen editierbar)				
		Vorgabe	Letzter Wert	
Variable	Bezeichnung	Vorgabe	Letzter W.	Neuer W.
dt	Zeitintervall	0.05	0.05	0.05
mt	Meßzeit	10	10	10
m	Masse in kg	10	10	10
k	Federkonst N/m	10	10	10
v	Anfangsgeschwindigkeit	0	0	0

Abb. 1.14

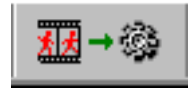


diese Einstellmöglichkeit hingewiesen, wenn es darum geht, geänderte oder neu erstellte Projekte abzuspeichern.

Bevor nun Änderungen vorgenommen werden können, die tiefer ins jeweilige Projekt eingreifen, ist vom bisher verwendeten Sofortstartmodus in den **Entwicklungsmodus** zu wechseln.

Erreicht wird dies mit Hilfe der in **Abb. 1.15** gezeigten Buttons.

Wechsel vom
Sofortstartmodus



zum



Entwicklungsmodus



Abb. 1.15

1.4.2 Änderungen am Kernprogramm vornehmen

Es kann vorkommen, dass Startwerte nicht im Startwerte-Fenster, sondern am Anfang des Kernprogramms deklariert werden. Sie können nur im Entwicklungsmodus dadurch geändert werden, dass die vorhandenen Zahlenwerte entfernt und entsprechend durch andere ersetzt werden. Das hierzu benötigte **Editor-Fenster** lässt sich öffnen, indem man aus dem Menüpunkt **Eingaben** den Unterpunkt **Programmeditor** aufruft bzw. die Taste **F5** drückt.

Änderungen im Startwerte-Fenster werden nicht berücksichtigt, wenn die vom Projekt benötigten Variablen zusätzlich noch im Kernprogramm definiert werden, was ein Programm-Fehler ist. Es ist also darauf zu achten, dass Variablendefinitionen nicht doppelt vorhanden sind.

Bevor man ein verändertes Projekt startet, ist es ratsam, das komplette Projekt unter einem neuen Namen abzuspeichern. Sollten nämlich unerwartet Probleme auftauchen, so gehen evtl. alle Änderungen verloren. An dieser Stelle sei nochmals erwähnt, dass man beim Speichern eines Projektes die Möglichkeit hat, dass sich das Projekt bei einem späteren Aufruf im Sofortstartmodus befindet, d.h. der physikalische Vorgang am Bildschirm kann mit Hilfe der Taste **Ablauf** sofort gestartet werden.



1.4.3 Variablen selbst definieren

Mit dem Unterpunkt **Variable definieren** des Menüpunktes **Eingaben** bietet PAKMA im Entwicklungsmodus die Möglichkeit, zusätzlich zu den Ausgabegrößen, weitere physikalische Größen für die Darstellung im Ausgaben-Fenster zu berechnen, ohne Veränderungen im Kernprogramm vornehmen zu müssen. Hier kann man solche Variablen definieren, die sich mit einfachen mathematischen Operationen aus den Ausgabegrößen sowie Konstanten des Projektes berechnen lassen. Für das Projekt **Federschwinger Fe-derschwingung.prj** könnte man beispielsweise die Spannenergie definieren, die sich aus der Auslenkung y und der Federkonstanten k ermitteln lässt. Nachdem man den Menüpunkt **Eingaben** → **Variablen definieren** → **Neue Variable definieren...** aufgerufen hat, wird das Fenster **Laufvariable definieren** (s. **Abb. 1.16**) geöffnet. In das Feld **Variablenname** trägt man nun die Bezeichnung der zu definierenden Größe (z. B. "E_span" für Energie), in das Feld daneben die zugehörige Berechnungsformel (z. B. " $0.5 \cdot k \cdot y \cdot y$ " für die Spannenergie der Feder) ein. Durch Anklicken der Schaltfläche **OK** bestätigt man die Eingabe und das Fenster wird wieder geschlossen.

Selbstdefinierte Variablen werden wieder gelöscht, indem der Menüpunkt **Eingaben** → **Variablen definieren** → **Selbstdefinierte Variable löschen** aufgerufen wird. Dazu wählt man durch Anklicken die Variable aus, die gelöscht werden soll. Es erscheint dann in der Dialogbox **Löschen** eine Abfrage, ob die ausgewählte Variable tatsächlich gelöscht werden soll. Nach entsprechender Wahl gelangt man wieder in das Fenster **Variable löschen**. Diesen Vorgang

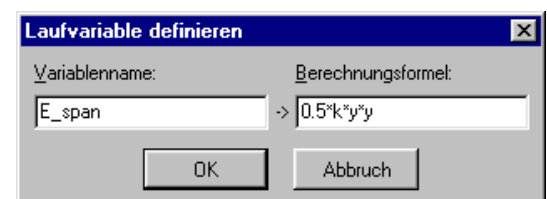


Abb. 1.16

kann man nun so lange wiederholen, bis alle selbstdefinierten Variablen gelöscht sind. Durch Anklicken der Schaltfläche **Fertig** wird das Fenster **Variable löschen** wieder geschlossen.

1.4.4 Ausgabe neu definierter Variablen

In PAKMA ist der Menüpunkt **Ausgaben** das zentrale Werkzeug, um Variablen in einem Ausgaben-Fenster darzustellen. Da hierzu genauere Kenntnisse nötig sind, werden im Folgenden die benötigten Unterpunkte detailliert beschrieben.

Folgende Anzeigeformen stehen im PAKMA zur Verfügung:

Darstellung einer Ausgabegröße als Zahlenwert (Unterpunkt **Großanzeige**)

Darstellung des Wertes einer Ausgabegröße in einem Messinstrument durch einen Zeiger (Unterpunkt **Zeigerinstrument**)

Darstellung einer Ausgabegröße durch einen veränderlichen Balken (Unterpunkt **Balkendarstellung**)

Darstellung von Ausgabegrößen in einem Diagramm (Unterpunkt **Graphendarstellung**)

Zu jeder dieser Darstellungsformen werden zugehörige Eigenschaften in separaten Fenstern festgelegt.

Die einzelnen Ausgabeformen werden ausführlicher im nächsten Kapitel vorgestellt, wenn es darum geht, selbst ein Projekt zu erstellen.



Damit die veränderten bzw. neu definierten Größen im Ausgaben-Fenster korrekt dargestellt werden, müssen evtl. die **Bereichsgrenzen** dieser Größen geändert werden.

1.4.5 Bereichsgrenzen nach Änderungen anpassen

Lässt man das Projekt mit den vorgenommenen Veränderungen ablaufen, kann es vorkommen, dass die Bereiche, in denen die Größen im Ausgaben-Fenster graphisch dargestellt werden, den neuen Werten nicht mehr angepasst sind. Dies ist beispielsweise dann der Fall, wenn die Achse eines Diagramms durch den eingezeichneten Graphen nicht mehr optimal ausgenutzt wird. Der erwähnte Bereich wird durch die sogenannten **Bereichsgrenzen** bestimmt, die in dem Fenster **Bereichsgrenzenliste** festgelegt sind und dort verändert werden können. (s. **Abb. 1.17**)

In **Kapitel 2** wird die **Bereichsgrenzenliste** nochmals behandelt. Daher sollen an dieser Stelle nur die hier benötigten Schritte kurz aufgeführt werden.

Nach dem Definieren neuer Variablen, wird über den Menüpunkt **Eingaben** der Unterpunkt **Bereichsgrenzen** aufgerufen. Wie man dort erkennen kann, befinden sich die neuen Variablen bereits in dieser Liste.

Man muss jetzt nur noch die zugehörigen leeren Felder ausfüllen. In die Spalte **links/unten** bzw. **rechts/oben** sind die minimal bzw. maximal möglichen Werte der neuen Variable einzugeben.

Die Spalte **Bezeichnung** sollte ebenfalls den zugehörigen Eintrag erhalten.

Zum Übernehmen der Daten muss jetzt nur noch das Fenster geschlossen werden.

Variable	Referenz	links/unten	rechts/oben	Bezeichnung
t		0	mt	t/s
y		-20	20	y/cm
v	y			v/cm/s
a	y			a/m/s ²

Abb. 1.17

Projekte, bei denen Messungen durchgeführt wurden, können bei der Reproduktion nur bezüglich der Auswertung verändert werden, da die Vorgaben



des Experiments natürlich nicht mehr abzuändern sind. Häufig sind jedoch von verschiedenen Versuchsdurchführungen mehrere Reihen von Messdaten in einem solchen Projekt abgespeichert, so dass sich entsprechend der aufgerufenen Messdaten unterschiedliche Reproduktionen ergeben.

Selbstverständlich lassen sich zusätzliche Variablen aus den Ausgabegrößen des Projektes berechnen, welche dann in den Ausgaben-Fenstern graphisch dargestellt werden können.

Abschließend soll wieder an das Speichern, insbesondere an die hierbei bestehende Möglichkeit des anschließenden Startens im Sofortstartmodus erinnert werden.



2 Erstellen von Projekten

In einem fertigen Projekt werden Module zu einer Projektdatei, d. h. zu einer Datei mit der Erweiterung ***.prj** zusammengefasst. Jedes einzelne dieser Module lässt sich unabhängig vom eigentlichen Projekt abspeichern, und kann anschließend von einem anderen Projekt wieder aufgerufen werden. Diese Möglichkeit erleichtert die Arbeit, wenn mit Hilfe vorhandener Module neue Projekte erstellt bzw. schon vorhandene verändert werden sollen. Die Moduldateien sind anhand ihrer Dateierweiterungen zu erkennen, welche in folgender Aufzählung in Klammer angegeben sind.

Im Allgemeinen besteht ein PAKMA-Projekt aus folgenden Teilen:

- dem **Kernprogramm** (*.pkp)
- einem oder mehreren **Ausgaben-Fenstern** (*.out)
- den **Bereichsgrenzen** für die Ausgabegrößen (*.dat)
- den aufgenommenen **Messdaten** (*.mdt).

Prinzipiell unterscheidet man zwei Formen von PAKMA-Projekten:

„**Experimental**“-**Projekte** mit Messdatenerfassung, mit deren Hilfe physikalische Messungen durchgeführt werden können bzw. durchgeführt wurden, so dass mit den mitabgespeicherten Messwerten der Ablauf und die Auswertung der jeweiligen Experimente reproduzierbar sind.

Modelle, d. h. Projekte, die ohne Messungen physikalische Sachverhalte präsentieren.

Werden in einem Projekt physikalische Abläufe unter Einbezug relevanter physikalischer Größen animiert dargestellt, so wird dies als **dynamisch ikonische Repräsentation** bezeichnet. Diese Animationen können in PAKMA in Echtzeit, d. h. parallel zu einem Experiment, dessen Messdaten mit PAKMA aufgenommen werden, ablaufen.

Vorgehensweise beim Erstellen eines Projektes:

- **physikalisches Modell konzipieren bzw. mit VisEdit erstellen**
- **entsprechendes Kernprogramm schreiben bzw. mit VisEdit automatisch erstellen lassen**
- **Startwerte in Menü eintragen**
- **Bereichsgrenzen festlegen**
- **Ausgaben-Fenster gestalten**
- **Projekt speichern**
- **Programmlauf durchführen**
- **Animationen einfügen und mit den Ausgabegrößen verknüpfen**
- **endgültiges Speichern**
- **erneuter Programmlauf**

Bevor an Hand des schon bekannten Beispiels des Federschwingers gezeigt wird, wie man selbst ein Projekt erstellen kann, werden in den folgenden Abschnitten die hierfür nötigen Grundlagen erläutert.

2.1 Das Kernprogramm

Der zentrale Bestandteil eines jeden Projekts ist das Kernprogramm. Es dient dazu, Messanweisungen auszuführen – dies wird in Kapitel 4 erläutert – und/oder ein physikalisches Modell

durch mathematische Operationen und bestimmte Befehle zu beschreiben. Beim Compilieren wird mit Hilfe eines Parsers das erstellte Kernprogramm in eine exe- Datei übersetzt, die der jeweilige Rechner ausführen kann. Die berechneten, bzw. gemessenen Werte können anschließend in den PAKMA Ausgaben-Fenstern ausgegeben werden.

Eine Alternative zum selbsterstellten Kernprogramm ist das Erstellen eines Wirkungsgefüges mit dem graphischen Editor VisEdit. Dies wird in Kapitel 3 kurz vorgestellt.



Hinweis

In diesem Abschnitt sollen nun die Richtlinien gegeben werden, nach denen ein Kernprogramm zu erstellen ist. Ein Kernprogramm besteht im wesentlichen aus Wertzuweisungen und mathematischen Operationen, aus einfachen Schleifenstrukturen und aus speziellen PAKMA-Befehlen.

Zusätzlich zu dem im Kernprogramm verwendeten Quelltext können Funktionen und Prozeduren, die mit Hilfe von PASCAL erstellt wurden, so verwendet werden, wie dies in dieser Programmiersprache üblich ist. Da im Folgenden nicht die komplette Programmiersprache PASCAL behandelt werden kann und soll, wird an dieser Stelle auf die reichlich vorhandene Fachliteratur verwiesen.



Hinweis

Nach dem Starten von PAKMA ist bereits ein leeres Projekt geöffnet. Außerdem befindet sich PAKMA bereits im Entwicklungsmodus. Dies lässt sich an den entsprechenden Icons in der Symbolleiste erkennen (s. Abb. 2.1).



Abb. 2.1

Wie bereits erwähnt, wird das Kernprogramm in das Fenster **Editor** eingegeben. Man öffnet das Editorfenster, indem man aus dem Menüpunkt **Eingaben** den Unterpunkt **Programmeditor** aufruft bzw. die Taste **F5** drückt. Dabei stellt man fest, dass sich die Menüleiste bei geöffnetem und aktivem Editor-Fenster verändert hat.

Bereits in **Kapitel 1** wurden einige Menüpunkte und deren Unterpunkte ausführlich erläutert. Nun sollen die Veränderungen in der Menüleiste, die bei geöffnetem und aktivem Editor-Fenster auftreten, besprochen werden. Während die Menüpunkte **Eingaben**, **Ausgaben**, **Prog-Lauf**

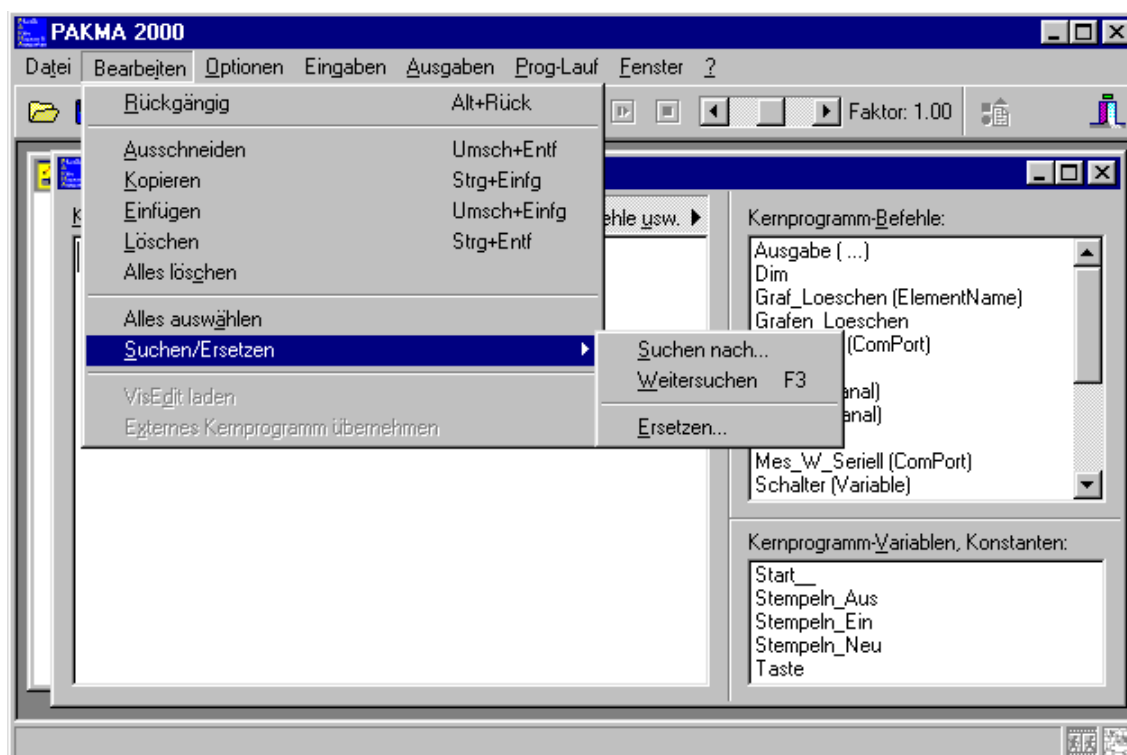


Abb. 2.2

und **Fenster** mit ihren Unterpunkten gleich geblieben sind, haben sich die Unterpunkte des Menüpunktes **Bearbeiten** verändert und dienen nun der Bearbeitung des Editor-Fensters. Außerdem ist ein Menüpunkt **Optionen** hinzugekommen.

2.1.1 Der Menüpunkt Bearbeiten

Mit Hilfe der Unterpunkte des Menüpunktes Bearbeiten (s. **Abb. 2.2**) kann man den Text im **Editor**, also das Kernprogramm, bearbeiten.

Rückgängig /Alt+Backspace/

Dieser Befehl macht die jeweils zuletzt durchgeführte Veränderung (dies kann eine Eingabe oder auch ein Löschen sein) im Fenster **Editor** rückgängig.

Es kann immer nur eine, nämlich die zuletzt vorgenommene Veränderung rückgängig gemacht werden.



Ausschneiden /Strg+X/

Der markierte Text bzw. die markierten Zeichen werden in die Zwischenablage des Computers verschoben und dabei im aktuellen Fenster gelöscht. In der Zwischenablage werden sie gespeichert und können später wieder in das Editor-Fenster eingefügt werden (s. Befehl **Einfügen**).

Kopieren /Strg+C/

Im Gegensatz zum Unterpunkt **Ausschneiden** wird mit diesem Befehl der durch Anklicken markierte Text in die Zwischenablage des Rechners kopiert, ohne aus dem aktuellen Editor-Fenster entfernt zu werden.

Einfügen /Strg+V/

Mit diesem Befehl wird der Inhalt der Zwischenablage in das Editor-Fenster an die Stelle eingefügt, an der sich der Cursor zur Zeit befindet.

Löschen /Strg+X/

Damit wird der markierte Text aus dem Editor-Fenster gelöscht.

Alles löschen /Strg+Entf/

Mit diesem Befehl wird der gesamte Inhalt des Editor-Fensters, also das gesamte Kernprogramm, gelöscht.

Suchen

Ruft man diesen Punkt auf, erscheint ein Untermenü, dessen Befehle zum **Suchen** und Ersetzen eines Textes im Editor-Fenster dienen.

Suchen → Suchen nach ...

Hiermit kann nach bestimmten Begriffen, Variablen oder sonstigen Zeichenketten im Text des Editor-Fensters gesucht werden. Bei Anklicken dieses Unterpunktes wird zunächst das Fenster **Suchen** (s. **Abb. 2.3**) geöffnet.

Nun gibt man in das Feld für **Suchen nach** den gewünschten Text ein. Zusätzlich besteht die Möglichkeit, beim Suchen auf Groß- und Kleinschreibung zu achten. Dies geschieht durch Anklicken des dafür vorgesehenen Kontrollkästchens. Das Suchen beginnt immer an der

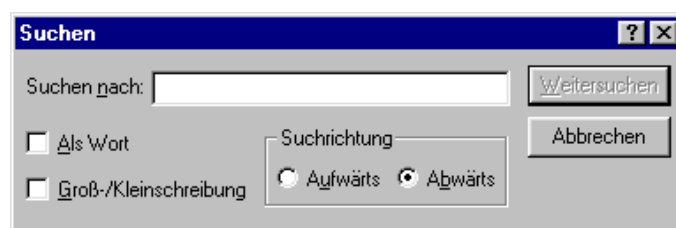


Abb. 2.3

Stelle, an der sich der Cursor gerade befindet, und endet entweder, wenn der gesuchte Text gefunden wurde, oder wenn das Ende des Kernprogramms im Editor-Fenster erreicht wurde. Die Fundstelle erscheint farbig unterlegt, wobei man durch den jeweiligen Text scrollen muss, um an diese Stelle zu gelangen. Wurde der gesuchte Text bis zum Ende des Kernprogramms nicht gefunden, erfolgt ein Warnton als entsprechende Rückmeldung.

Suchen → Ersetzen ...

Dieser Befehl dient dazu, einen Text im Editor zu suchen und durch einen anderen zu ersetzen. In den entsprechenden Eingabefeldern bestimmt man, welche Begriffe (bzw. welche Zeichenketten) gesucht und durch andere, ebenfalls einzutragende, Begriffe ersetzt werden sollen. Zusätzlich stehen bestimmte Optionen zur Verfügung, die jeweils durch Anklicken der Kontrollkästchen aktiviert werden können. Klickt man Groß/Kleinschreibung an, so wird beim Suchen auf Groß- und Kleinschreibung geachtet. Ist Alle aktiviert, so werden alle im Text des Editor-Fensters vorkommenden gesuchten Zeichen ersetzt. Das Ersetzen endet dann am Ende des Kernprogramms. Andernfalls endet das Ersetzen sobald die gesuchten Zeichen einmal gefunden und ersetzt wurden. Wenn die Option Mit Bestätigung aktiviert ist, erfolgt jedesmal, wenn die gesuchte Zeichenkette gefunden wurde, eine erneute Abfrage, ob die gesuchten Zeichen tatsächlich ersetzt werden sollen. Die drei Optionen können außerdem beliebig miteinander gekoppelt werden.

Suchen → Weitersuchen /F3/

Dieser Befehl wird im Zusammenhang mit dem Befehl **Suchen nach ...** verwendet. Wie bereits erläutert, endet **Suchen nach ...**, sobald der Begriff einmal gefunden wurde. Mit **Weitersuchen** kann man die Suche nach dem gewünschten Begriff im noch verbleibenden Text fortführen.

2.1.2 Der Menüpunkt Optionen

Die Unterpunkte des Menüpunktes **Optionen** (s. **Abb. 2.4**) dienen dazu, das Schreiben eines Kernprogramms erheblich zu vereinfachen. PAKMA ist prinzipiell in der Lage, gewisse Formalien, die bei der Eingabe eines Kernprogramms zu beachten sind, automatisch zu erledigen. Die Überprüfung des Kernprogramms auf diese Formalien wird durch die Unterpunkte des Menüpunktes **Optionen** eingestellt und aufgerufen.

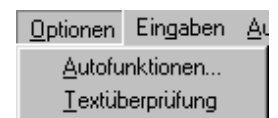


Abb. 2.4

Autofunktionen ...

Beim Anklicken dieses Unterpunktes wird das Fenster **Automatische Funktionen** (s. **Abb. 2.5**) geöffnet. In diesem Fenster kann man festlegen, welche Formalien bei der Eingabe eines Kernprogramms automatisch von PAKMA erledigt werden sollen. Das Aktivieren einer Funktion geschieht durch Anklicken des entsprechenden Kontrollkästchens.

Die in diesem Fenster aktivierten Funktionen werden während der Eingabe des Kernprogramms beim Verlassen jeder Zeile automatisch überprüft.

Automatisches Einrücken

Diese Funktion dient dem Strukturieren des Kernprogramms und beeinflusst nur die äußere Form. Der Text wird damit z. B. innerhalb einer Schleife automatisch gleichmäßig eingerückt.

Automatische Zeilenkontrolle

Ist diese Funktion aktiviert, erscheinen bei verschiedenen Eingabefehlern Meldungen auf dem Bildschirm. Wurde beispielsweise innerhalb einer Zeile eine Klammer geöffnet und nicht wieder geschlossen, werden Sie bei Verlassen der Zeile darauf aufmerksam gemacht. Eine Meldung erscheint auch dann, wenn Fehler in der Syntax, also in der Schreibweise, aufgetreten sind.

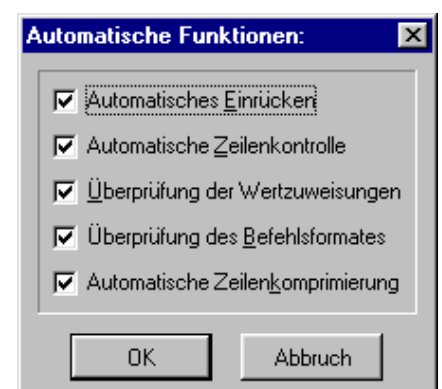


Abb. 2.5

Überprüfen der Wertzuweisungen

In Turbo-Pascal haben Wertzuweisungen immer die Form $a := \dots$. Wurde diese Funktion aktiviert, muss bei der Eingabe nicht darauf geachtet werden, dass der Doppelpunkt vorhanden ist, da er automatisch von PAKMA beim Verlassen einer Zeile hinzugefügt wird.

Überprüfen des Befehlsformates

In Turbo-Pascal ist es notwendig, dass am Ende einer Zeile ein Semikolon steht. Wenn die Funktion **Überprüfen** des Befehlsformates aktiviert ist, wird das Semikolon beim Verlassen einer Zeile automatisch von PAKMA hinzugefügt, sofern es noch nicht vorhanden ist.

Automatische Zeilenkomprimierung

Ist diese Funktion aktiviert, werden innerhalb einer Zeile des Kernprogramms automatisch nur die Leerzeichen zugelassen, die in Turbo-Pascal notwendig sind.

Textüberprüfung

Wenn dieser Unterpunkt aufgerufen wird, wird der gesamte bisher eingegebene Text des Kernprogramms auf die im Fenster **Automatische Funktionen** ausgewählten Optionen überprüft und gegebenenfalls korrigiert.

2.1.3 Der Aufbau eines Kernprogramms

In einem Kernprogramm müssen zu Beginn immer Startwerte und Konstanten definiert werden. Anders als in einem Pascal-Programm übernimmt PAKMA aber intern die Deklaration der Variablen. Wenn keine anderen Anweisungen getroffen werden, werden alle Variablen dem Typ Real (reelle Zahlen) zugeordnet.

Der übrige Teil besteht aus einer Schleifenstruktur (z. B. Repeat-Until-Schleife), in welcher die iterative Berechnung der gesuchten Größen durchgeführt wird. Dabei werden die Ausgabegrößen i. allg. schrittweise in Abhängigkeit einer weiteren Größe „z“ (z. B. der Zeit oder des Weges) berechnet. Die Größe „z“ wird dann für den folgenden Schleifendurchlauf um die als Konstante festzulegende Schrittweite „dz“ erhöht. Die Schleife wird so oft durchlaufen, bis ein vorher festgelegter maximaler Wert der Größe „z“ überschritten wird. Durch diesen maximalen Wert wird der „Umfang“ des Modells bestimmt.

Damit die gesuchten Größen in einem Ausgaben-Fenster ausgegeben und veranschaulicht werden können, müssen ihre Werte jeweils nach einer Berechnung zur Verfügung gestellt (ausgegeben) werden. Hierfür existiert folgender wohl wichtigster PAKMA-Befehl, ohne den kein PAKMA-Projekt Informationen nach außen abgeben kann:

ausgabe (a,b,c,...)

Bei den Größen a,b,c,..., die als Argument in dieser „Ausgabe“-Anweisung stehen, handelt es sich um die Größen, die in den Ausgaben-Fenstern eines Projektes veranschaulicht werden sollen.

In der Ausgabeanweisung müssen alle Größen aufgeführt sein, die in den Ausgaben-Fenstern eines Projektes dargestellt werden sollen.

Ein Kernprogramm hat also folgenden prinzipiellen Aufbau:

Als Größe „z“ wurde hier die Zeit t gewählt.




```
t:= 0 ; ... ;      //Festlegen der Startwerte und Konstanten
dt:= ... ;        //Festlegen der Schrittweite
mt:= ... ;        //Festlegen der Ablaufzeit mt
REPEAT            //Schleifenbeginn
  Wertzuweisungen; //Berechnung der Ausgabegrößen
  und Rechenbefehle; //jeweils für einen Wert von t
  ausgabe (Variablenliste); //Ausgabeanweisung
  t:=t+dt ;        //Erhöhung von t um Schrittweite dt
UNTIL t > mt;      //Abbruchbedingung der Schleife
```

als in einem Pascal-Programm können innerhalb einer Zeile mehrere Anweisungen stehen, die jeweils durch ein Semikolon voneinander zu trennen sind. Weiterhin besteht die Möglichkeit, bei der Eingabe in jeder Zeile einen Kommentar einzufügen, der durch zwei aufeinanderfolgende Schrägstriche und sinnvollerweise durch Leerstellen von den eigentlichen Anweisungen getrennt werden muss.

*PAKMA ist in der Lage, gewisse Formalien zu erledigen, die sonst bei der Eingabe eines PASCAL-Programms beachtet werden müssten. Es ist z. B. nicht erforderlich die in Turbo-Pascal bei Wertzuweisungen notwendigen Doppelpunkte ($a:=...$) sowie das Semikolon am Zeilenende zu schreiben. Sie werden von PAKMA beim Verlassen jeder Anweisungszeile automatisch eingefügt werden. Wenn man z.B. „ $F = -k * y$ “ eingibt, so wird beim Verlassen der Zeile daraus „ $F := -k * y;$ “.*



Welche Möglichkeiten außerdem für die automatische Korrektur bei der Eingabe des Kernprogramms in PAKMA zur Verfügung stehen, wurde bereits zuvor behandelt.

Nachdem man das Kernprogramm komplett eingegeben hat, kann man das Editor-Fenster schließen.

*Falls man später bestimmte Anfangswerte oder Konstanten für verschiedene Durchläufe des Modells verändern will, so ist es wesentlich übersichtlicher und einfacher, diese nicht am Anfang des Kernprogramms zu definieren, sondern hierfür das Fenster **Startwerte** zu verwenden. Es ist jedoch darauf zu achten, dass die Werte nicht zweimal definiert werden.*



2.1.4 Das Startwerte-Fenster

Im Zusammenhang mit dem Startwerte-Fenster bietet PAKMA die Möglichkeit, den Umgang mit Anfangswerten und Konstanten übersichtlich zu gestalten. Wie bereits im vorangegangenen Abschnitt erläutert, können diese Größen am Anfang eines Kernprogramms definiert werden. PAKMA stellt außerdem mit dem Fenster **Startwerte** eine weitere Möglichkeit zur Verfügung. Diese ist dann von Vorteil, wenn man für verschiedene Abläufe des Projektes Anfangswerte oder Konstanten verändern möchte, um damit die Wertzuweisungen, die einen physikalischen Sachverhalt charakterisieren, besser überblicken zu können.

Man öffnet das Fenster Startwerte (s. Abb. 2.6), indem man aus dem Menüpunkt **Eingaben** den Unterpunkt **Startwerte** aufruft bzw. die Taste **F6** drückt. Es erscheint eine Tabelle mit folgenden fünf Spalten:

Variable

In dieser Spalte wird das im Kernprogramm verwendete Symbol der Größe eingetragen, deren Wert definiert werden soll (z. B.: „D“ für Federkonstante).

Bezeichnung

Um die Arbeit mit dem Startwerte-Fenster übersichtlich zu gestalten, wird hier die Bedeutung der jeweiligen Variablen eingetragen ggf. mit Angabe der gewählten physikalischen Einheit.

Vorgabe

Der hier eingegebene Wert ist ein vom Projektautor für diese Variable vorgeschlagener Zahlenwert, welcher für einen ersten Durchlauf gut geeignet ist. Hier sollte man anfänglich nichts verändern.

Letzter W.

Hier wird vom Programm automatisch der Wert der Variablen eingetragen, der beim letzten Ablauf des Projektes verwendet wurde. Auch hier sollte man ebenfalls nichts verändern.

Neuer W.

Der in dieser Spalte eingetragene Wert wird vom Programm für den nächsten Ablauf verwendet. Dieser Wert kann beliebig verändert werden. Man kann das jeweilige Eingabefeld beschriften, nachdem man dieses mit der linken Maustaste angeklickt hat. Es erscheint daraufhin ein blinkender Cursor in dem entsprechenden Feld. Durch Drücken der **Tabulator**-Taste gelangt man in das darauf folgende Feld. Die Werte werden nach dem Schließen des Fensters automatisch übernommen.

*In PAKMA 2000 ist es im Entwicklungsmodus möglich, an Stelle von Zahlen einfache mathematische Operationen in die Wertfelder einzutragen, z. B. $3*2$ oder $3/2$.*



Hinweis

Variable	Bezeichnung	Vorgabe	Letzter W.	Neuer W.
dt	Zeitintervall	0.05	0.05	0.05
mt	Meßzeit	10	10	10
m	Masse in kg	10	10	10
k	Federkonst N/m	10	10	10
v	Anfangsgeschwindigkeit	0	0	0

Abb. 2.6

2.2 Die Bereichsgrenzen

Damit man die Ausgabegrößen eines Projektes in den Ausgaben-Fenstern graphisch darstellen kann, muss zunächst für jede dieser Größen ein Bereich festgelegt werden, innerhalb dessen die Darstellung erfolgen soll. Dieser Bereich wird durch die sogenannten **Bereichsgrenzen** angegeben.

Die Bereichsgrenzen werden im Fenster **Bereichsgrenzenliste** festgelegt. Man öffnet es, indem man aus dem Menüpunkt **Eingaben** den Unterpunkt **Bereichsgrenzen** aufruft bzw. die Taste **F7** drückt. Ist das Fenster **Bereichsgrenzenliste** geöffnet und aktiv, dann hat sich die Menüleiste von PAKMA geändert. Durch Anklicken stellt man fest, dass sich die Unterpunkte des Menüpunktes **Bear-**

Variable	Referenz	links/unten	rechts/oben	Bezeichnung
t		0	mt	t/s
y		-20	20	y/cm
v	y			v/cm/s
a	y			a/m/s ²

Abb. 2.7

beiten verändert haben. Weiterhin ist ein Menüpunkt **Optionen** mit verschiedenen Unterpunkten hinzugekommen.

2.2.1 Der Menüpunkt Bearbeiten

Die Unterpunkte des Menüpunktes **Bearbeiten** (s. Abb. 2.8) dienen dazu, bestimmte inhaltliche Veränderungen und Eintragungen im Fenster **Bereichsgrenzenliste** vorzunehmen.

Im Einzelnen bedeuten die Unterpunkte Folgendes:

Variablenliste bereinigen

Im Fenster **Bereichsgrenzenliste** werden in der Spalte **Variable** vom Programm automatisch alle Variablen aufgelistet, die in der Ausgabeanweisung des Kernprogramms aufgeführt sind. Wurden in diesem Befehl eine oder mehrere Variablen gelöscht, um Veränderungen an dem Projekt vorzunehmen, so werden diese erst beim Aufruf des Befehls **Variablenliste bereinigen** aus der Spalte **Variable** entfernt.

Der Befehl **Variablenliste bereinigen** dient nur zum Löschen von "alten" Ausgabevariablen. Es werden keine neuen Variablen hinzugefügt. Dies geschieht automatisch, sobald das Fenster **Bereichsgrenzenliste** geöffnet wird.

Die nun folgenden Unterpunkte **alle Variablen Autoskalieren** sowie **einzelne Variablen Autoskalieren** beziehen sich auf die Möglichkeit, die Bereichsgrenzen von Variablen automatisch vom Programm bestimmen zu lassen.

alle Variablen Autoskalieren

Bei Aufruf dieses Menüpunktes werden die Bereichsgrenzen von PAKMA automatisch für alle in der Spalte **Variable** aufgelisteten Größen festgelegt. Diesen Befehl können Sie allerdings erst aufrufen, wenn das Kernprogramm bereits einmal abgelaufen ist, da dem Rechner vorher noch keine Zahlenwerte für die einzelnen Größen zur Verfügung stehen. Das Programm bestimmt nun selbstständig den maximalen und minimalen Wert einer Ausgabegröße und setzt diese Werte um etwa 10% vergrößert als Bereichsgrenzen ein.

einzelne Variablen Autoskalieren...

Mit diesem Befehl kann man die Bereichsgrenzen für einzelne Variablen automatisch vom Programm festlegen lassen. Auch diesen Unterpunkt kann man erst verwenden, wenn das Kernprogramm bereits abgelaufen ist. Nachdem man den Befehl aufgerufen hat, wird das Fenster **Autoskalieren** geöffnet.

In diesem Fenster werden alle Ausgabegrößen untereinander aufgeführt (z.B.: t, a, v, y). Nachdem man durch Anklicken die Variable, welche automatisch skaliert werden soll, ausgewählt hat, erscheint vor der Variable ein Pfeil. Auf diese Art und Weise kann man mehrere Größen nacheinander auswählen. Durch Anklicken der Schaltfläche **OK** bestätigt man die Auswahl, und das Fenster **Autoskalieren** wird wieder geschlossen. PAKMA führt nun die Autoskalierung für die ausgewählten Größen durch und die Bereichsgrenzen werden im Fenster **Bereichsgrenzenliste** in die entsprechenden Felder eingetragen.

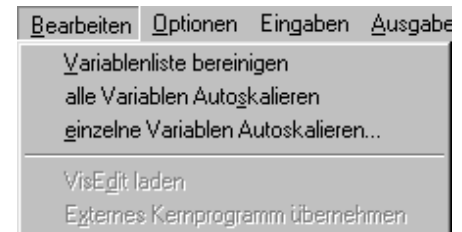


Abb. 2.8

2.2.2 Für Sonderfälle: Optionen

Die Unterpunkte des Menüpunktes **Optionen** (s. Abb. 2.9) werden nur selten nötig sein. Sie dienen dazu, den Umgang mit den Bereichsgrenzen noch vielseitiger zu machen. Man kann für jedes Projekt drei verschiedene Datensätze von Bereichsgrenzen für die Ausgabegrößen eingeben und nutzen, wodurch man die Bereichsgrenzen zwischen zwei Abläufen wechseln kann,

ohne sie neu eingeben zu müssen. Dies ist dann hilfreich, wenn Konstanten oder Anfangswerte zwischen zwei Durchläufen verändert werden sollen und die alten Bereichsgrenzen für den neuen Durchlauf nicht mehr brauchbar sind.

Ein typisches Beispiel ist die Analyse der Schwingung eines physikalischen Pendels bei kleiner bzw. großer Anfangsauslenkung.

Die drei Datensätze sind durch die Nummerierungen Datensatz 1, 2 bzw. 3 gekennzeichnet. Zusätzlich können den Datensätzen Namen zugewiesen werden; ohne besondere Zuweisung lauten sie: **Standard**, **Alternative 1** und **Alternative 2**. In der Kopfzeile des Fensters **Bereichsgrenzenliste** kann man erkennen, welcher Datensatz gerade ausgewählt ist.

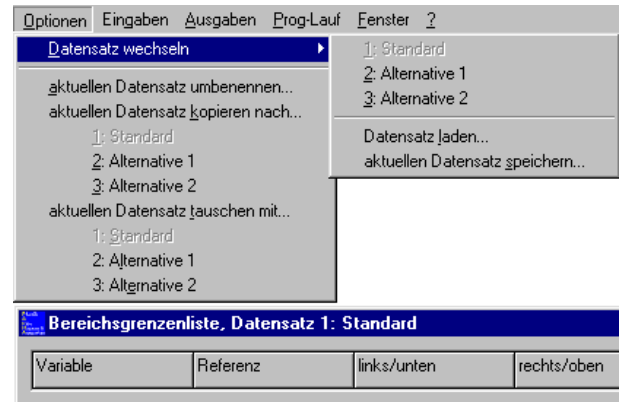


Abb. 2.9

Datensatz wechseln

Mit diesem Unterpunkt kann man den aktuellen Datensatz gegen einen der beiden anderen austauschen. Man wendet ihn an, wenn ein Projekt beim nächsten Durchlauf mit einem anderen Datensatz von Bereichsgrenzen ablaufen soll bzw., wenn man einen zusätzlichen Datensatz neu definieren möchte. Unter dem Befehl **Datensatz wechseln** befindet sich ein weiteres Untermenü mit den Punkten **1:Standard**, **2:Alternative 1** und **3:Alternative 2**, wovon man jeweils nur zwei auswählen kann. Der gerade aktuelle Datensatz ist dabei nicht aufrufbar. Wenn man den gewünschten Datensatz durch Anklicken ausgewählt hat, erscheint dieser im Fenster **Bereichsgrenzenliste**.

Ferner findet man unter **Datensatz wechseln** die Unterpunkte **Datensatz laden...** und **aktuellen Datensatz speichern...**, die dazu dienen einen einzelnen Datensatz auf der Festplatte bzw. einem Diskettenlaufwerk zu speichern oder von dort in den Arbeitsspeicher zu laden. Diese Befehle sind nicht zu verwechseln mit den Befehlen **Datei → Modul laden/speichern → Bereichsgrenzen**, bei denen jeweils drei Datensätze gleichzeitig geladen bzw. gespeichert werden.

aktuellen Datensatz umbenennen

Mit diesem Befehl kann man dem aktuellen Datensatz einen neuen Namen zuweisen. Bei Aufrufen dieses Unterpunktes wird das Fenster **Umbenennen** (s. Abb. 2.10) geöffnet.

In diesem Fenster können Sie in das Feld **Neuer Name** den gewünschten Namen für den aktuellen Datensatz eingeben. Im Feld **Alter Name** erscheint automatisch der bisherige Name des aktuellen Datensatzes.

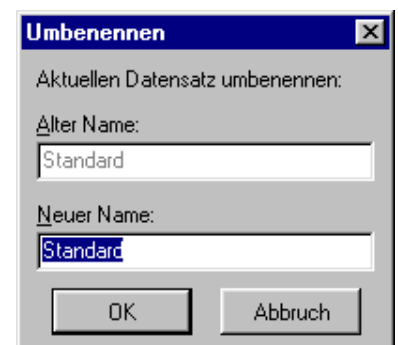


Abb. 2.10

Es ist sinnvoll, solche Namen einzutragen, die den Umgang mit den Datensätzen erleichtern. Hat man z. B. beim physikalischen Pendel unterschiedliche Anfangsauslenkungen gewählt, bietet es sich an, die Bereichsgrenzen zum einen „große Auslenkung“ und zum anderen „kleine Auslenkung“ zu nennen.



aktuellen Datensatz kopieren nach...

Dieser Befehl dient dazu, den Inhalt des aktuellen Datensatzes in einen anderen Datensatz zu kopieren, der dadurch überschrieben wird. Der aktuelle Datensatz wird dabei nicht verändert.

aktuellen Datensatz tauschen mit...

Mit diesem Befehl können Sie den aktuellen Datensatz gegen einen anderen austauschen, um die Nummerierung von mehreren bereits erstellten Datensätzen sinnvoll zu ändern.

2.3 PAKMA Anweisungen

Im Anhang zu diesem Handbuch befindet sich eine ausführliche Liste der PAKMA-Anweisungen und spezieller PAKMA-Funktionen einschließlich zugehöriger Kurzbeschreibung.

Außerdem kann man eine nicht kommentierte Liste von Anweisungen und Konstanten direkt vom Editorfenster aus öffnen (s. Abb. 2.11). Diese Liste hat den besonderen Vorteil, dass man dort aufgeführte Anweisungen etc. durch Doppelklick direkt ins Kernprogramm übernehmen kann.

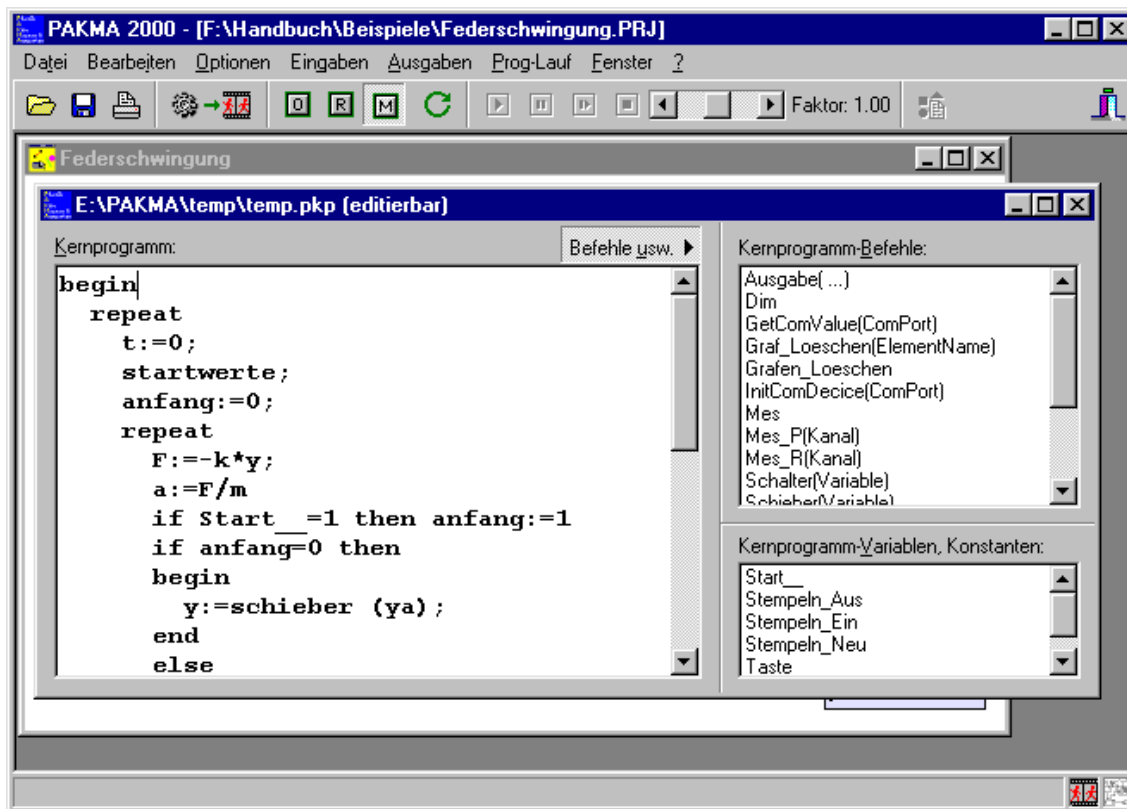


Abb. 2.11

Im Folgenden werden häufig verwendete Befehle aufgeführt, die auch im nächsten Abschnitt benötigt werden, um das bereits erwähnte Beispielprojekt erstellen zu können. Prinzipiell sind jedoch gewisse Programmierkenntnisse bei der Verwendung dieser Befehle von Vorteil.

Repeat ... Until

Hiermit wird eine Schleife erstellt, die solange durchlaufen wird, bis eine oder mehrere Abbruchbedingungen erfüllt sind.

For ... to ... do

Mit diesem Befehl lassen sich ebenfalls Schleifen konstruieren, die jedoch nach einer bestimmten Anzahl von Durchläufen beendet werden.

If ... then ...

Für die Abfrage von bestimmten Werten wird dieser Befehl bereitgestellt.

Ausgabe (...)

Dieser Befehl sendet die Werte der in der Klammer angegebenen Variablen an die Ausgaben-Fenster. Damit jede Veränderung der Variablen auch vollzogen wird, muss sich dieser Befehl innerhalb derjenigen Schleifen befinden, in denen die benötigten Variablen berechnet werden. Es ist jedoch darauf zu achten, dass dieser Befehl nicht mehrmals in einem Schleifendurchlauf aufgerufen wird. Dies kann vor allem dann der Fall sein, wenn verschiedene Schleifen miteinander verknüpft sind.

Rechenoperationen

$+$, $-$, $*$, $/$ sind die üblichen Rechenfunktionen, zu denen **sin**, **cos**, **arctan**, **ln**, **sqr**, **sqrt**, **exp**, ... hinzukommen. Die Zahl π ist durch die Konstante **pi** mit maximaler Genauigkeit bereits vordefiniert.

Schieber, Schalter und Taster

Auf die jeweiligen Werte wird im Kernprogramm durch folgende Wertzuweisung **var := schieber(x)** zugegriffen. **var** ist hier eine Variable, die den Wert des Schiebers erhält, **schieber** kann gegebenenfalls durch **schalter** bzw. **taster** ersetzt werden. **x** ist diejenige Variable, die im **Eigenschaften-Dialog** des betreffenden Schiebers festgelegt wurde, s. Kap. 2.5.

2.4 Die Ausgabeelemente

2.4.1 Großanzeige

Bei dieser Darstellungsform wird während eines Programmlaufes eine Ausgabegröße als Zahlenwert im Ausgaben-Fenster angezeigt. Hierbei wird der Zahlenwert bei jedem Aufruf des Ausgabebefehls **ausgabe (...)** im Kernprogramm, d. h. bei jedem Schleifendurchlauf, aktualisiert.

Abb. 2.12 zeigt ein Beispiel für die Darstellung des Ortes als Zahlenwert im Projekt **Federschwingung**.

Soll eine Ausgabegröße auf diese Art und Weise ausgegeben werden, ist aus dem Menüpunkt **Ausgaben** der Unterpunkt **Großanzeige** aufzurufen. Es erscheint daraufhin das Fenster **Eigenschaften Großanzeige** (s. **Abb. 2.13**), dessen Eingabefelder folgende Bedeutung haben:

Variable

In dieses Feld wird die gewünschte Ausgabegröße eingegeben, die durch die Großanzeige dargestellt werden soll. Dies geschieht menügesteuert, indem man den Schalter (Pfeil) neben dem Eingabefeld anklickt und aus den aufgelisteten Ausgabegrößen die gewünschte Größe ebenfalls durch Anklicken auswählt.

Feldgröße

Hier gibt man eine Zahl ein, welche die Größe des Feldes bestimmt, das im Ausgaben-Fenster für den Zahlenwert inkl.

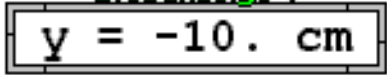
Großanzeige 1


Abb. 2.12

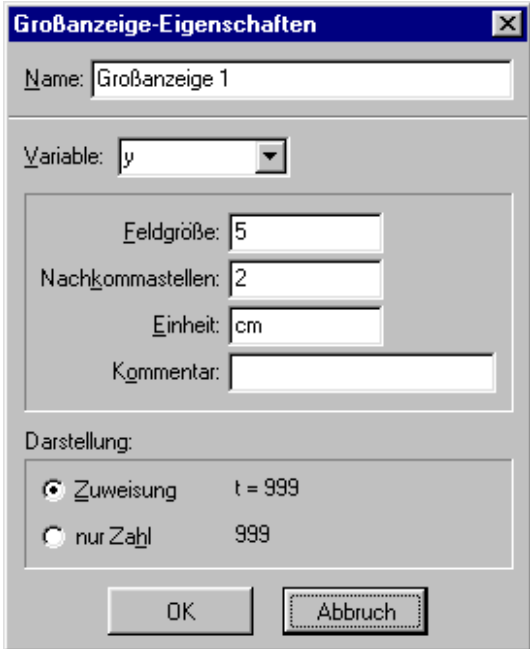


Abb. 2.13

Vorzeichen bereitgestellt wird. Die eingegebene Zahl gibt die gesamte Anzahl der Stellen an, welche zur Darstellung der Ausgabegröße benötigt werden. Die Anzahl der Nachkommastellen muss für die Gesamtstellenzahl mitgezählt werden.

Als Variable wurde die Auslenkung ausgewählt. Diese läuft während des Ablaufs des Projektes **Federschwingung** von -20 cm bis 20 cm. Als Genauigkeit soll hier für die Anzahl der Nachkommastellen 2 gewählt werden. Die Zahl, die man nun für Feldgröße einträgt, muss also mindestens 4 betragen. Sie kann jedoch auch größer sein. In unserem Beispiel wurde 5 eingetragen.

Nachkommastellen

Hier kann man, je nach gewünschter Genauigkeit, die Anzahl der Stellen festlegen, die nach dem Komma angezeigt werden sollen. Dazu trägt man die entsprechende Ziffer in das Kästchen ein. Für zwei Stellen nach dem Komma wurde eine 2 eingetragen.

Einheit

In dieses Eingabefeld kann man, wenn gewünscht, eine Einheit für die darzustellende Größe eintragen. Diese wird dann im Ausgaben-Fenster an den Zahlenwert im Abstand von einer Leerstelle angefügt. Für die Auslenkung wurde die Einheit cm eingetragen.

Kommentar

Hier kann ein Kommentar zur Ausgabegröße eingegeben werden, der dem Zahlenwert im Ausgaben-Fenster vorangestellt wird. Die gewünschten Leerstellen sind dabei mit einzutragen. Weiterhin kann man die Art der Darstellung des Zahlenwertes wählen. Hierbei stehen zwei Möglichkeiten zur Verfügung, die man durch Anklicken der entsprechenden Felder auswählt:

Der Zahlenwert kann mit Zuweisung dargestellt werden. Dabei erscheint die Variable der Ausgabegröße mit Gleichheitszeichen vor dem Zahlenwert im Ausgaben-Fenster.

Der Zahlenwert kann ohne Zuweisung (nur Zahl) dargestellt werden. In diesem Fall erscheint im Ausgaben-Fenster nur der Zahlenwert der Ausgabegröße sowie ggf. ein Kommentar und die Einheit.

Hat man die gewünschten Eigenschaften ausgewählt und eingetragen, bestätigt man die Eingaben durch Anklicken der Schaltfläche **OK**. Das Fenster **Großanzeige-Eigenschaften** ist geschlossen und man kehrt zum Ausgaben-Fenster zurück. Hier erscheint nun in einem Rahmen die gewählte Großanzeige der Ausgabegröße.

Der vorhandene Rahmen zeigt an, dass das Feld aktiviert ist. Es können nun Größe und Position des Feldes angepasst werden. Wie hierbei vorzugehen ist, soll im Folgenden erläutert werden. Diese Vorgehensweise gilt prinzipiell für alle Anzeigeformen der Ausgabegrößen.

Feld positionieren

Man kann das Feld im Ausgaben-Fenster an eine gewünschte Stelle setzen. Dazu klickt man die Fläche des Anzeigefeldes mit der Maus an und verschiebt es bei gedrückter linker Maustaste an die gewünschte Stelle. Die Maustaste darf erst dann wieder losgelassen werden, wenn die gewünschte Position erreicht ist.

Größe des Feldes verändern

Auch die Größe des Anzeigefeldes kann wie üblich verändert werden. Hierzu wird die Maus z. B. auf die rechte untere Ecke des Rahmens geführt, so dass sich der Mauszeiger als Doppelpfeil erscheint. Man klickt nun die linke Maustaste und zieht den Rahmen bei gedrückter Taste zur gewünschten Größe auf. Die Maustaste wird wiederum erst dann losgelassen, wenn die gewünschte Größe erreicht ist.

Feld deaktivieren

Um den Rahmen, der anzeigt, dass das Feld aktiviert ist, wieder auszublenden, klickt man im Ausgaben-Fenster an eine beliebige andere Stelle außerhalb des Rahmens, z. B. neben das Feld.

Feld erneut aktivieren

Um einen Rahmen zu aktivieren muss man generell zuvor den Editiermodus eingeschaltet haben. Dies geschieht mit Hilfe des Unterpunktes **Editieren** im Menüpunkt **Bearbeiten**. Möchte man das Feld nun aktivieren, muss man nur den Inhalt des Feldes einmal anklicken. Der Rahmen erscheint dann erneut.

Eigenschaften des Feldes wieder anzeigen

Möchte man die gewählten Eigenschaften für eine erstellte Großanzeige bzw. für eine andere Darstellungsform erneut einsehen, um ggf. Veränderungen vorzunehmen (z. B. die Zahl der Nachkommastellen), so muss man auf den Inhalt des Feldes (z.B. die **Großanzeige**) doppelklicken. Das zugehörige Fenster **Eigenschaften...** (z. B. **Großanzeige- Eigenschaften**) wird daraufhin geöffnet.

*Um die Eigenschaften für ein bereits vorhandenes Feld erneut anzeigen zu lassen, darf aus dem Menüpunkt **Ausgaben** nicht der entsprechende Unterpunkt aufgerufen werden, da dieser Aufruf nur eine weitere Großanzeige erzeugt.*



Farbe wählen

Die Farbe, mit der die Großanzeige auf dem Bildschirm dargestellt wird, wird bei der vorliegenden Version noch durch die Einstellung unter dem Menüpunkt: Animationselemente -> linien-farbe festgelegt, s. Abb. 2.23, und ist dort zu ändern.

Soll eine Anzeigeform, z. B. eine **Großanzeige**, wieder aus dem aktuellen Ausgaben-Fenster entfernt werden, muss man diese zunächst durch Anklicken aktivieren. Anschließend klickt man auf die entsprechenden Symbole in der Symbolleiste oder man ruft aus dem Menüpunkt **Bearbeiten** den Unterpunkt **Ausschneiden** auf.

2.4.2 Zeigerinstrument

Um den Wert einer Ausgabegröße im Ausgaben-Fenster darzustellen, kann man statt der digitalen Großanzeige auch ein Messgerät mit Zeiger wählen. Der Ausschlag des Zeigers ändert sich dann bei jedem Aufruf der Ausgabeanweisung im Kernprogramm mit dem Wert der Ausgabegröße.

Abb. 2.14 zeigt ein Beispiel für die Darstellung einer Ausgabegröße in einem Zeigerinstrument.

Möchte man ein Zeigerinstrument erstellen, ruft man aus dem Menüpunkt **Ausgaben** den Unterpunkt Zeigerinstrument auf. Es wird das Fenster **Eigenschaften Zeigerinstrument** geöffnet (s. **Abb. 2.15**).

Man kann hier im Feld **Variable** menügesteuert auswählen, welche der Ausgabegrößen des Projektes im Zeigerinstrument dargestellt werden soll. Nachdem man den Schalter (Pfeil) rechts neben dem Eingabefeld angeklickt hat, werden die Ausgabegrößen untereinander aufgelistet. Die Auswahl erfolgt nun durch Anklicken der gewünschten Größe.

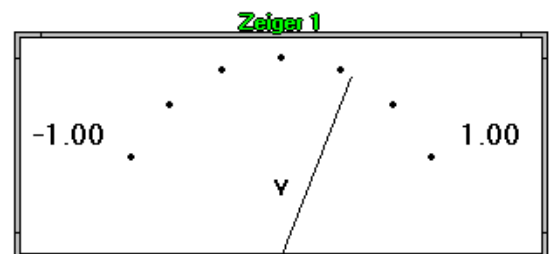


Abb. 2.14

Durch Anklicken der Schaltfläche **OK** bestätigt man den Eintrag und das Fenster **Eigenschaften Zeigerinstrument** wird wieder geschlossen. Im Ausgaben-Fenster erscheint nun das Zeigerinstrument in einem Rahmen. Die Größe und Position des Feldes kann nun ebenso verändert werden, wie dies zuvor für die Großanzeige beschrieben wurde.

2.4.3 Balkendarstellung

Neben den bisher beschriebenen Darstellungsformen kann man eine Ausgabegröße auch durch einen an einer Skala veränderlichen Balken veranschaulichen. Die nebenstehende **Abb. 2.16** zeigt ein Beispiel für diese Anzeigeform.

Nachdem man aus dem Menüpunkt **Ausgaben** den Unterpunkt **Balkendarstellung** aufgerufen hat, erscheint das Fenster **Eigenschaften Balkendarstellung** (s. **Abb. 2.17**), in dem die Form der Balkendarstellung festgelegt wird.

Im Feld **Variable** kann man menügesteuert die gewünschte Ausgabegröße, z. B. **t** für die Zeit, auswählen. Dazu klickt man auf den Schalter (Pfeil) rechts neben dem Eingabefeld, woraufhin alle Ausgabegrößen untereinander aufgelistet werden. Die gewünschte Ausgabegröße wählt man nun durch Anklicken aus.

Anschließend kann man entscheiden, ob der Balken in horizontaler oder vertikaler Richtung im Ausgaben-Fenster dargestellt werden soll. Dazu klickt man im Feld **Ausrichtung horizontaler Balken** bzw. **vertikaler Balken** an. Im entsprechenden Kontrollfeld, z. B. vor **horizontaler Balken**, erscheint ein schwarzer Punkt.

Auch die **Balkenfarbe** und die **Achsenfarbe** können menügesteuert festgelegt werden, indem man die Schalter rechts neben den Eingabefeldern anklickt und die gewünschte Farbe auswählt. Wird als Achsenfarbe **unsichtbar** gewählt, so erhält man einen Balken ohne Skalierung. Entsprechend kann eine Skala ohne Balken dargestellt werden, indem als Balkenfarbe **unsichtbar** gewählt wird.

Zusätzlich befinden sich im Fenster **Eigenschaften Balkendarstellung** folgende Optionen, die sich jeweils durch Anklicken aktivieren lassen:

Überlauf sichtbar

Geht eine Balkendarstellung beim Ablauf des Projektes über das Skalenende hinaus, so wird dies durch einen Farbwechsel sichtbar gemacht. Da die Skala des Balkens durch die Bereichsgrenzen festgesetzt wird, ist es in einem solchen Fall empfehlenswert, die zugehörigen Bereichsgrenzen zu verändern.

Farbwechsel wenn negativ

Ist diese Option aktiviert, so ändert der Balken jeweils beim Durchgang durch die Nulllage seine Farbe.

Durch Anklicken der Schaltfläche **OK** bestätigt man die Eingaben und verlässt das Fenster **Eigenschaften Balkendarstellung**. Im aktuellen Ausgaben-Fenster erscheint dann die definierte Balkendarstellung in einem Rahmen, der anzeigt, dass die Balkendarstellung aktiviert ist. Man kann nun das Anzeigenfeld im Ausgaben-Fenster an die gewünschte Stelle positionieren und



Abb. 2.15

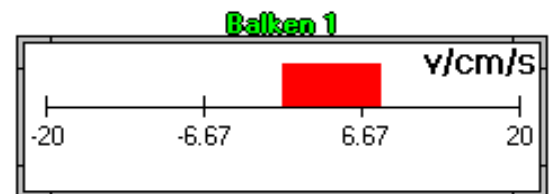


Abb. 2.16

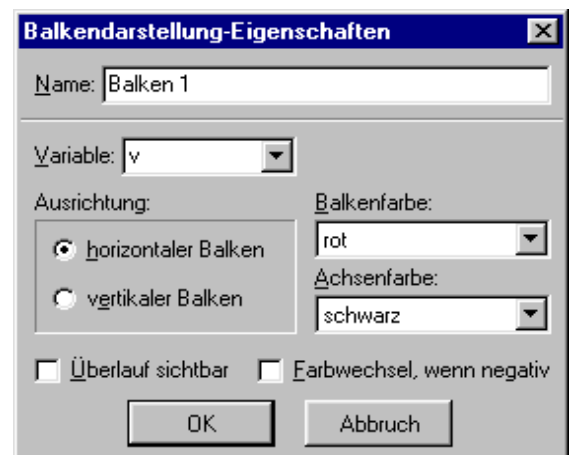


Abb. 2.17

zur gewünschten Größe aufziehen. Dabei kann sowohl die Breite als auch die Länge des Balkens verändert werden.

2.4.4 Graphendarstellung

Mit den jeweils beschriebenen Anzeigeformen kann man eine einzelne Ausgabegröße in einem Ausgaben-Fenster darstellen. Soll jedoch eine funktionelle Abhängigkeit mehrerer Ausgabegrößen voneinander untersucht werden, bietet es sich an, diese in einem Diagramm darzustellen. Dazu ruft man aus dem Menüpunkt **Ausgaben** den Unterpunkt **Graphendarstellung** auf. Dann erscheint das Fenster **Graphendarstellung-Eigenschaften** (s. Abb. 2.18). In diesem Fenster können die Eigenschaften für eine graphische Darstellung in einem Diagramm festgelegt werden.

Zunächst muss jedoch zu jedem Projekt eine Graphenliste erstellt werden (sofern dies noch nicht geschehen ist). Aus dieser sind diejenigen Graphen auszuwählen, die in einem Diagramm dargestellt werden sollen. Zum Erstellen der Graphenliste klickt man im Fenster Graphendarstellung-Eigenschaften die Schaltfläche **Neu** an. Daraufhin erscheint eine neue Zeile im Eigenschaftsmenü, in der jeweils ein Graph charakterisiert werden kann.

In dieser Zeile legt man in einer Tabelle mit den Spalten x-Achse, y-Achse, Typ und Farbe alle Einstellungen für den zu zeichnenden Graphen fest, wobei mehrere Graphen (die der Reihe nach jeweils mittels Klick auf **Neu** festgelegt werden müssen) in ein oder mehrere Diagrammen gezeichnet werden können. Die Eingabe in die entsprechenden Felder erfolgt menügesteuert, indem man jeweils den Schalter (Pfeil) rechts neben dem entsprechenden Eingabefeld anklickt. Den gewünschten Eintrag wählt man dann durch Anklicken aus.

Die Eintragungen in den Spalten sind folgendermaßen vorzunehmen:

x-Achse

Nachdem man den Schalter (Pfeil) rechts neben dem Feld angeklickt hat, werden untereinander alle Ausgabegrößen aufgelistet. Nun wählt man diejenige Ausgabegröße durch Anklicken aus, die für einen Graphen in horizontaler Richtung in einem Diagramm aufgetragen werden soll.

y-Achse

Hier wird diejenige Ausgabegröße eingetragen, die für den Graphen in vertikaler Richtung im Diagramm aufgetragen werden soll.

Typ

Hier kann man wählen, auf welche Art und Weise die Werte im Diagramm dargestellt werden sollen. Dazu stehen **Dreiecke**, **Kreuze**, **Punkte**, **Linien**, **L.m.P.(Linie mit kleinen Kreisen)**, **Stäbe**, **Kreise** und **Rechtecke** zur Verfügung, die bei Anklicken des Schalters aufgelistet werden.



Abb. 2.18



Farbe

Hier wird die Farbe ausgewählt, in der der Graph gezeichnet werden soll.

Man legt nun untereinander alle gewünschten Graphen in jeweils einer Zeile fest.

*Auch wenn verschiedene Diagramme in einem oder mehreren Ausgaben-Fenstern dargestellt werden sollen, werden alle Graphen in einer gemeinsamen Graphenliste definiert. Es gibt also zu jedem Projekt genau **eine** Graphenliste.*



Nun können die Eigenschaften für jeweils ein Diagramm ausgewählt werden. Mit Hilfe eines Klicks auf die Linke Spalte (☒) werden die aufgeführten Graphen ausgewählt. *Ausgewählte Graphen* erkennt man daran, dass *vor der Zuordnung des Graphen ein kleiner Haken* erscheint. Sollen einzelne Graphen wieder deaktiviert werden, klickt man den jeweiligen Graphen erneut an. Das Häkchen ist dann verschwunden. Zusätzlich kann eine einzelne Zeile über den Button **Löschen**  **Löschen** bzw. die komplette Graphenliste über **Alles Löschen**  **Alles löschen** gelöscht werden.

Weitere Einstellungsmöglichkeiten:

Autoskalierung

Die hier einstellbaren Optionen sind nur in Sonderfällen relevant. Stellt man nach dem Ablauf eines Projektes fest, dass eine Achse eines Diagramms aufgrund der festgelegten Bereichsgrenzen nicht optimal ausgenutzt wird (Beispiel: Die Achse ist von -3 bis 3 skaliert, der Graph verläuft jedoch nur von -1 bis 1), kann man diese Achse mit Hilfe der **Autoskalierung** neu einteilen lassen.

Bei einer **Autoskalierung** ermittelt der Rechner den maximalen und minimalen Wert, den die Ausgabegröße während des letzten Programmablaufs angenommen hat. Diese Werte werden dann, um etwa 10% vergrößert, als Randwerte für die neue Skalierung der Achse verwendet.

Durch Anklicken der Optionen **Rechtswertachse** bzw. **Hochwertachse** kann man entscheiden, ob die horizontale Achse, die vertikale Achse oder beide vom Programm autoskaliert werden sollen. Wurden an einer Achse mehrere Ausgabegrößen dargestellt, so erfolgt die Autoskalierung für alle aufgetragenen Größen. Ggf. erscheinen dann mehrere verschiedene Achseneinteilungen im Diagramm.

Eine Autoskalierung wird im Diagramm wirksam, sobald die Schaltfläche **OK** angeklickt wird. Das Projekt muss nicht noch einmal gestartet werden.

Die Autoskalierung der Achsen eines Diagramms erfolgt prinzipiell auf die gleich Art, wie die Autoskalierung der Bereichsgrenzen. Der wesentliche Unterschied besteht jedoch darin, dass bei Autoskalierung der Achsen eines Diagramms die Bereichsgrenzen im Fenster Bereichsgrenzenliste nicht verändert werden, d. h. Sie können die alten Bereichsgrenzen leicht wieder zurückrufen, indem Sie die Autoskalierung wieder ausschalten.



Graphengröße:

Mit Hilfe der beiden Listenfelder **Breite** und **Höhe** kann die genaue Graphengröße in Pixeln festgelegt werden.

Hat man im Fenster **Graphendarstellung-Eigenschaften** alle gewünschten Einträge vorgenommen, so kann man diese mit **OK** bestätigen. Im Ausgaben-Fenster erscheint dann das zusammengestellte Diagramm innerhalb eines Rahmens, der anzeigt, dass das Diagramm aktiviert ist.

Nun kann man das Diagramm bei gedrückter linker Maustaste an die gewünschte Stelle im Ausgaben-Fenster positionieren sowie zur gewünschten Größe aufziehen (falls die Graphen-

größe nicht schon im Eigenschaftsmenü eingestellt wurde). Dies erfolgt auf die gleiche Weise, wie bei den anderen Darstellungsformen. Um das Feld zu deaktivieren, erneut zu aktivieren sowie die Eigenschaften des Feldes wieder anzeigen zu lassen, geht man ebenfalls wie zuvor beschrieben vor.

2.5 Schieber, Schalter und Taster

Die komfortabelste Form, die Größe von Variablen bzw. Startwerte vorzugeben bzw. zu verändern, stellen die sog. **Schieber** dar. Mit ihrer Hilfe kann man sogar während eines Programmlaufes den Wert einer Variablen ändern und damit Einfluss auf den dargestellten physikalischen Vorgang nehmen.

Schalter bieten die Möglichkeit zwischen zwei bestimmten Werten hin und her zu schalten.

Taster springen bei Betätigung vom gewählten Wert direkt zum Ausgangswert zurück.

In diesem Abschnitt soll dargelegt werden, wie solche **Schieber**, **Schalter** und **Taster** in ein Programm zu implementieren sind.

Man gelangt zu diesen Funktionen über den Menüpunkt **Eingaben** (s. Abb. 2.20). Nach entsprechender Wahl, wird im Ausgaben-Fenster das jeweilige Eingabe-Element zur Verfügung gestellt. Doch zuvor erscheint der zugehörige **Eigenschaften-Dialog**.

Als **Eingabe-Variable** kann man einen beliebigen Namen wählen, über den man dann im Kernprogramm auf die Werte des Schieber zugreifen kann.

Als **Name im Fenster** ist der Name anzugeben mit dem der Schieber im Ausgaben-Fenster bezeichnet werden soll.

Der Unterpunkt **Einheit** ermöglicht die Eingabe der entsprechenden physikalischen Bezeichnung für den eingestellten Zahlenwert.

Minimum und **Maximum** geben den Eingabebereich an, der durch die **Anzahl der Stufen** (= Anzahl der Werte - 1) entsprechend unterteilt ist.

Durch Anklicken kann man des Weiteren die Art der Teilung bestimmen. Ebenso lässt sich die Darstellungsform und die Verfügbarkeit, ob der Schieber immer (d.h. während des gesamten Projektablaufs) oder nur vor dem Ablauf (d.h. nur solange wie die Ablauf-Taste inaktiv, also noch nicht betätigt ist) verfügbar/aktiv ist, angeben. Das ist für die Benutzerführung hilfreich, denn somit können Schieber, die nur der Einstellung der Startbedingungen dienen, auch nur in der Vorbereitungsphase, also bei nicht gedrückter Ablauf-Taste, betätigt werden.

Nach Betätigen des Buttons **OK**, ist im Ausgaben-Fenster der jeweilige Schieber zu sehen.

Zu einem späteren Zeitpunkt kann man diesen Dialog dadurch aufrufen, dass man zunächst den entsprechenden Schieber aktiviert. Dazu klickt man im aktivierten Editiermodus den Schieber mit der linken Maustaste an und es erscheint ein Rahmen, der den Schieber umgibt (s. Abb. 2.19). Nun muss man nur noch mit der linken Maustaste im Inneren dieses Rahmens doppelklicken.

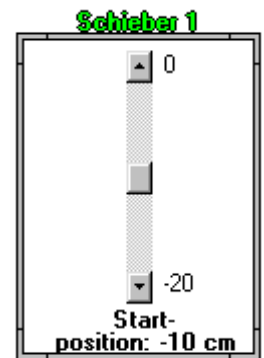


Abb. 2.19

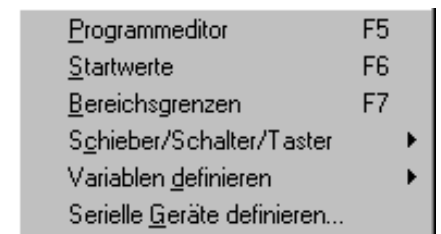


Abb. 2.20



Abb. 2.21

2.6.1 Das Rechteck

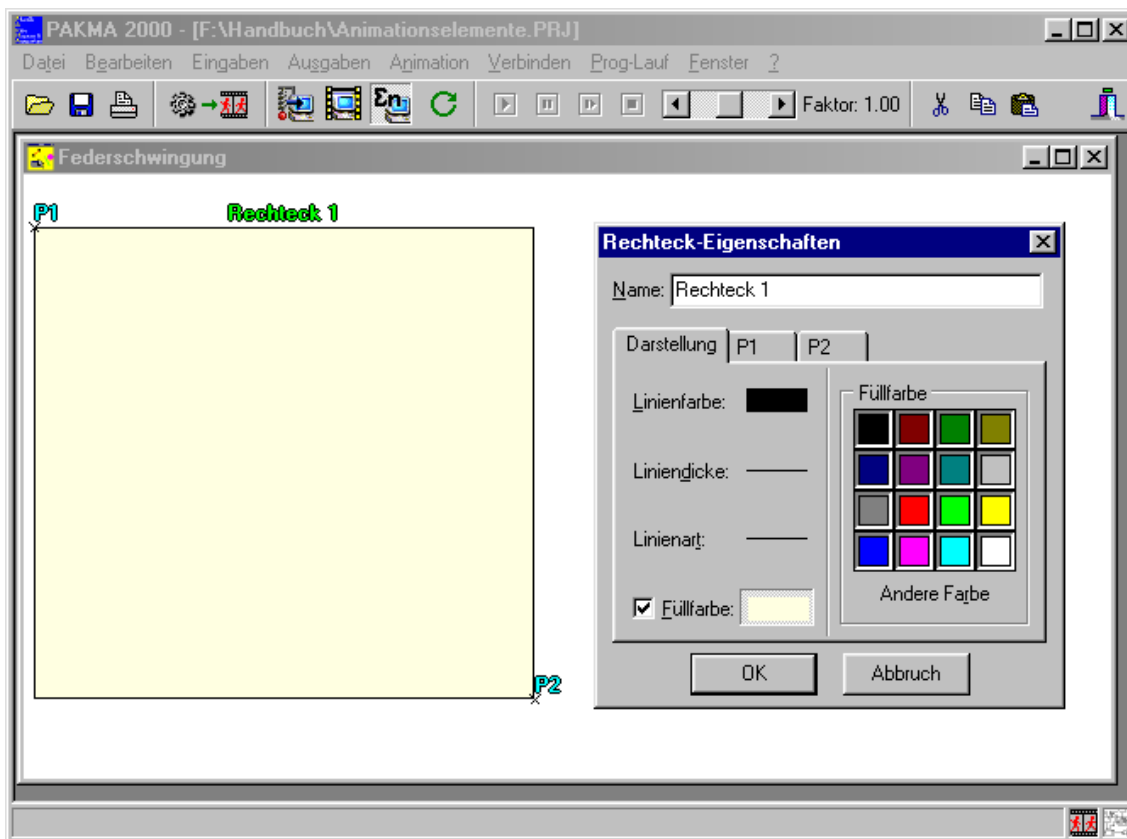


Abb. 2.24

Abb. 2.24 zeigt den Eigenschaften-Dialog des Rechteckes, welcher drei verschiedene Registerkarten enthält, mit denen man sämtliche Eigenschaften des Rechteckes festlegen kann.

Es gibt Registerkarten, die für die spezifische **Design-Darstellung** des Animationselementes verantwortlich sind, mit deren Hilfe man die **Linienfarbe**, die **Linienstärke**, die **Linienart** und die **Füllfarbe** bestimmen kann, so wie man dies obiger Abbildung entnehmen kann.

Auf der linken Seite dieser Abbildung kann man auch das aufgezeichnete Rechteck erkennen, das durch die zwei Punkte **P1** und **P2** bestimmt ist. Jedes Animationselement hat ein bis drei aus solcher charakteristischen Punkte, denen jeweils eine eigene Registerkarte zugeordnet ist, mit der die Bewegung des jeweiligen Punktes und damit die Bewegung und Form des Animationselementes eindeutig festgelegt wird. Zur Auswahl stehen hierbei **fixiert**, **relativ**, **absolut**, **konstante Form** und **Nachläufer**, deren besondere Funktion nun erklärt wird.

Wählt man die Registerkarte **P1** aus, so ist zunächst als Voreinstellung der Unterpunkt **fixiert** aktiviert. Dies bedeutet, das Animationselement bleibt unverändert, also fixiert, an der Stelle im Ausgaben-Fenster, wo es nach dem Aufziehen platziert wurde. Man hat dann nur noch die Möglichkeit zu entscheiden, ob das Animationselement in den Hintergrund gezeichnet wird oder nicht. Ist diese Option aktiviert, so kann man die Zeichenebene des Elements nicht beeinflussen, d.h. andere Elemente liegen evtl. darüber und können es somit verdecken. Der Punkt **P2** ist ebenfalls als **fixiert** festgelegt.

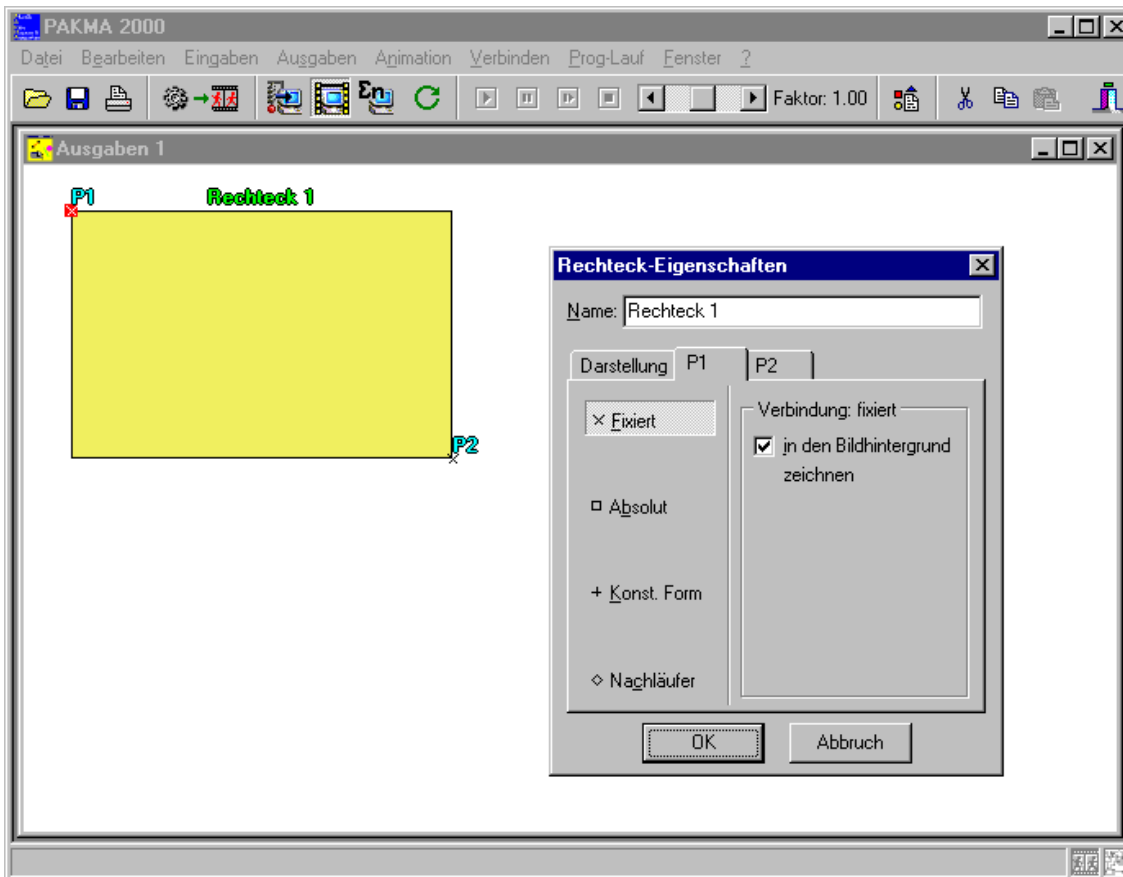


Abb. 2.25

Die wohl am häufigsten verwendete Einstellungsoption ist **absolut**. Hier können der x- bzw. y-Koordinate des jeweiligen Punktes Werte von beliebigen Ausgabegrößen zugeordnet werden (s. Abb. 2.26). Der Position des Punktes im Fenster wird demnach der Wert einer Ausgabegröße hinzugefügt, wobei man die hierfür benötigten Grenzen auswählen kann. Zum einen sind dies die festgelegten Bereichsgrenzen multipliziert mit dem ebenfalls einstellbaren Faktor, zum anderen, falls ein Graph vorhanden, die hier ausgegebenen maximalen Graphenwerte. Durch Festlegen der Position des zu Grunde liegenden Nullpunktes, ist die Bewegung des Punktes P1 eindeutig bestimmt.

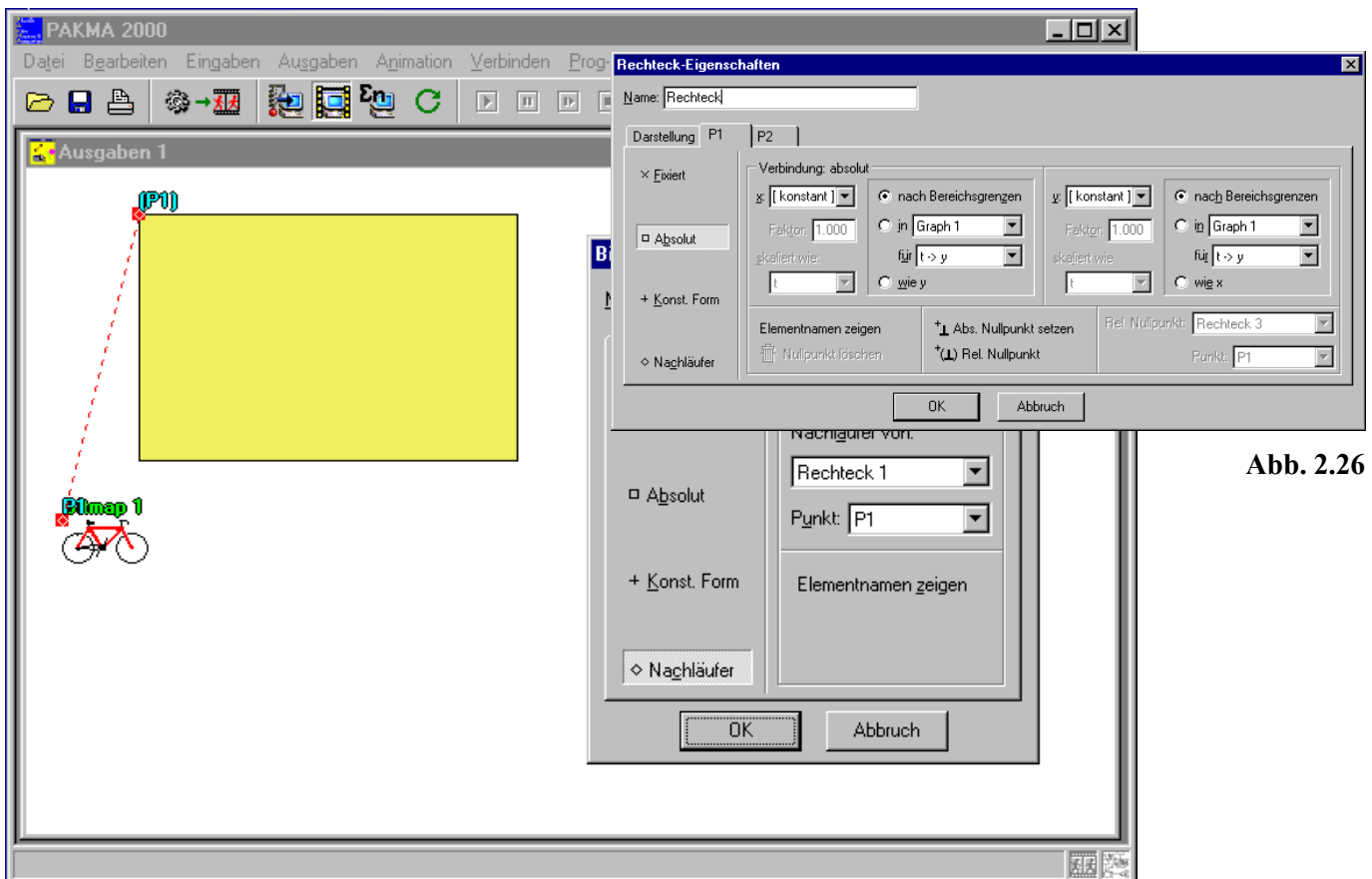


Abb. 2.26

Abb. 2.27

Der Unterpunkt **Nachläufer** ist dann zu wählen wenn sich der Markierungspunkt wie der Markierungspunkt eines anderen Animationselements bewegen soll, d. h. ihm im momentan gegebenen x, y-Abstand nachfolgen soll. In dem hier gezeigten Fall wird festgelegt, dass der Punkt P1 des Bitmaps dem Punkt P1 des Rechtecks, dem zuvor eine Bewegung zugeordnet wurde, in festem Abstand folgen soll. Diese Einstellung wird sofort durch eine gestrichelte Verbindungslinie im Ausgaben-Fenster visualisiert (s. **Abb. 2.27**).

*Beim Wechsel der Bewegungsoptionen gehen bereits neu vorgenommene Einstellungen verloren, da zunächst immer die Voreinstellungen präsentiert werden. Erst durch Drücken des Buttons **OK** sind die Bewegungen der Punkte **P1** und **P2** eindeutig festgelegt.*

2.6.2 Das Dreieck

Das Animationselement **Dreieck** ist ein Rechteck-Animationselement, das längs der Diagonalen halbiert ist. Alle Gestaltungen bezüglich Farbe, Linienart sowie die Beschreibungen durch die beiden Markierungspunkte P1 und P2 entsprechen genau dem Rechteck-Animationselement, die in Kapitel 2.6.1 beschrieben werden. Beim ersten Plazieren des Dreiecks kann man im sich öffnenden Eigenschaftsdialog über die Karte Dreieck die Lage der Dreiecksspitze wählen. (s. **Abb. 2.28**). Während des Projektablaufs wird die Lage natürlich durch die Vorgaben für P1 und P2 bestimmt.

Da sich die Voreinstellungen des Animationselementes Vektor ein wenig von den gerade beschriebenen unterscheiden, wird nun dieses Element vorgestellt.

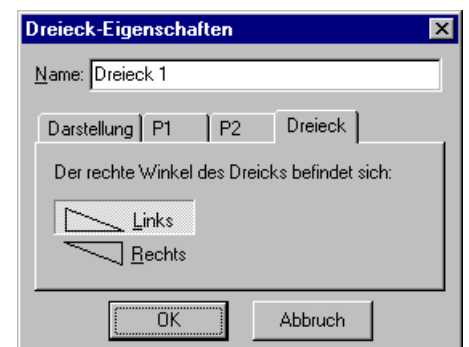


Abb. 2.28

2.6.3 Der Vektor

Nachdem man den Unterpunkt **Vektor** gewählt und das Element positioniert und auf die gewünschte Größe aufgezoogen hat, öffnet sich zunächst wieder der Dialog **Eigenschaften** (s. Abb. 2.30).



Abb. 2.29

Die Registerkarte **Darstellung** hat die schon zuvor beschriebene Funktion und legt allgemein das Aussehen fest, die Registerkarten **P1** und **P2** hingegen das jeweilige Bewegungsverhalten.

Es gibt jedoch einen wichtigen Unterschied bei den Voreinstellungen des Punktes **P2** im Vergleich zum Rechteck. Beim Unterpunkt **absolut** der Registerkarte **P2** wird automatisch der Punkt 1 als relativer Nullpunkt rechts unten eingetragen. Natürlich kann man diese Voreinstellung ändern und seinen individuellen Bedürfnissen anpassen.

Die letzte Registerkarte **Vektor** ist für das spezifische Aussehen des Animationselementes verantwortlich. Hier kann man einstellen, ob der Vektor einen Fuß besitzen soll und ob er in Doppellinien mit Füllfarbe gezeichnet werden soll. Die Linienstärke und Form wird weiterhin auf der 1. Darstellungskarte gewählt.

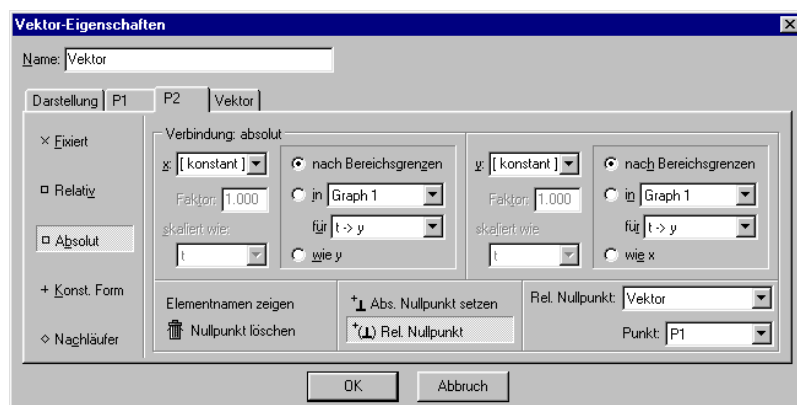


Abb. 2.30



Abb. 2.31

2.6.4 Breitpfeil

Nachdem man den Unterpunkt **Breitpfeil** des Menüpunktes **Animation** aufgerufen und den Breitpfeil in eine Richtung aufgezoogen hat erscheint das Fenster **Breitpfeil-Eigenschaften**. (s. Abb. 2.32)

Bei einem Breitpfeil handelt es sich um ein Animationselement, dessen Breite sich entsprechend des Wertes einer Ausgabegröße verändert, und das beim Ablauf des Projektes eine bestimmte Winkelstellung einnehmen kann. Der Drehwinkel (hier: α) wird über die Einträge in den Feldern **x-Komponente** und **y-Komponente** festgelegt.

Es gilt natürlich folgender mathematischer Zusammenhang:

$$\tan \nabla = \frac{y}{x}$$

Der Drehwinkel wird dabei von der positiven x-Achse ($\alpha = 0^\circ$) in mathematisch positivem Drehsinn gemessen.

In die Felder **x-Komponente** bzw. **y-Komponente** kann man die Eintragungen menügesteuert vornehmen, indem man den Schalter (Pfeil) rechts neben dem jeweiligen Eingabefeld anklickt. Dann hat man die Auswahl zwischen allen Ausgabegrößen des Projektes sowie dem Eintrag **[konstant]**. Durch die vorgenommenen Eingaben ist also die Richtung des Breitpfeils zu jedem Zeitpunkt **t** bei Ablauf des Projektes eindeutig bestimmt.

Die Breite des Elementes Breitpfeil ist beim Ablauf des Projektes immer an eine Ausgabegröße gebunden, welche im Feld **Breite gebunden an** menügesteuert eingegeben wird.

Außerdem kann man jeweils **Faktoren** einstellen, die bei der Drehung das Verhältnis von **x** zu **y** entsprechend verändern und somit Einfluss auf den Drehwinkel α haben. Analog dazu kann durch einen Faktor die Breite des Breitpfeils entsprechend dem eingegebenen Wert verändert werden.

Die Länge eines Breitpfeils bleibt beim Ablauf des Projektes immer konstant. Sie wird im Eingabefeld **Länge** in Pixel-Anzahl festgelegt, die man mit den Pfeilen vergrößern bzw. verkleinern kann. Man kann in dieses Feld auch schreiben, nachdem man es mit der Maus angeklickt hat.

Auch der **Drehpunkt** für die Bewegung (Drehung) ist festlegbar. Hierbei kann zwischen dem Mittelpunkt und dem Anfangspunkt des Breitpfeils gewählt werden. Die Auswahl erfolgt durch Anklicken des entsprechenden Kreises mit der Maus. In dem angeklickten Kreis erscheint dann ein schwarzer Punkt.

Durch Anklicken der Schaltfläche **OK** bestätigt man die vorgenommenen Eintragungen und das Fenster **Breitpfeil-Eigenschaften** wird geschlossen. Der erstellte Breitpfeil erscheint mit seinen beiden Markierungspunkten im Ausgaben-Fenster des Projektes. Dabei ist einer der Markierungspunkte ein Kreuz. Dieser symbolisiert den Drehpunkt der Bewegung und bleibt beim Ablauf des Projektes im Ausgaben-Fenster fixiert, sofern über ihn nicht noch zusätzlich auf der Registerkarte **P1** verfügt wird. Der zweite Markierungspunkt, der durch die Registerkarte **P2/Breitpfeil** festgelegt ist, ist ein Viereck. Dieser Punkt wird um den Drehpunkt gedreht.

Den zweiten Markierungspunkt (Viereck) kann man zusätzlich verschieben. Nur für den Fall, dass in einem der Felder **x-Komponente** oder **y-Komponente** der Eintrag **[konstant]** gewählt wurde, kann man durch die Lage von **P2** die spätere Drehbewegung beeinflussen. Wurde beispielsweise für die **x-Komponente** der Eintrag **[konstant]** gewählt und vergrößert man nun den Abstand zwischen Kreuz und Viereck, so ändert sich der Drehwinkel α nach der Berechnungsformel $\tan \alpha = y/x$, indem für **x** jetzt einen größeren Wert eingesetzt wird. Um den Markierungspunkt zu verschieben, führt man die Maus auf das Viereck, so dass aus dem Mauszeiger ein Doppelpfeil wird. Nun kann man bei gedrückter linker Maustaste den Abstand des Vierecks vom Drehpunkt (Kreuz) verändern.

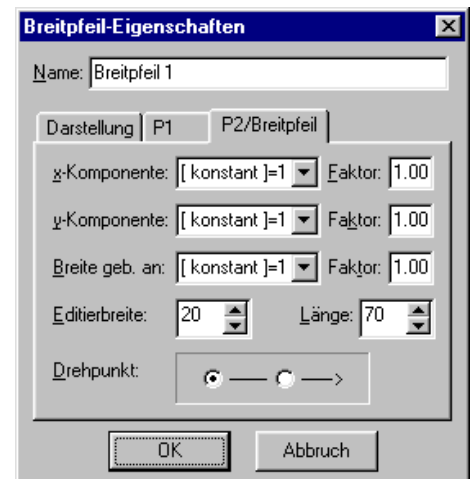


Abb. 2.32



□

Abb. 2.33



Abb. 2.34

2.6.5 Feder

Nachdem man den Anfangspunkt der Feder in das Ausgaben-Fenster positioniert und bei gedrückter linker Maustaste zur gewünschten Länge und in die vorgesehene Richtung aufgezogen hat, wird das Fenster **Feder-Eigenschaften** (s. Abb. 2.35) geöffnet.

Die spezifischen Eigenschaften der Feder, nämlich die Form können durch Eintragungen in der Registerkarte **Feder** mit den beiden Eingabefeldern **Breite** und **Anzahl Striche** festgelegt werden. Mit der Zahl im Feld **Breite** bestimmt man die Breite der Feder im Ausgaben-Fenster. Die Zahl im Feld **Anzahl Striche** gibt an, aus wie vielen Teilstrichen die Feder bestehen soll.



Abb. 2.35

2.6.6 Bitmap

Wurde als Animationselement ein Bitmap ausgewählt, so hat der Cursor jetzt die Gestalt eines Fahrrades. Nachdem das Fahrrad im Ausgaben-Fenster platziert wurde (die Größe ist dabei nicht veränderbar), erscheint das Fenster **Bildauswahl**. Hier werden verschiedene Bitmaps gezeigt, welche in das Ausgaben-Fenster eingefügt werden können, s. Abb. 2.37.

Bevor man sich endgültig für ein Bitmap entscheidet, kann man auswählen, ob das Bild **horizontal** oder **vertikal gespiegelt** werden soll. Abschließend wird das Bitmap per Mausklick in das Ausgaben-Fenster übertragen. Gleichzeitig wird das Fenster **Bildauswahl** geschlossen.

Das ausgewählte Bitmap erscheint durch ein Kreuz markiert im Ausgaben-Fenster. Es kann nun bei gedrückter linker Maustaste an die gewünschte Position verschoben werden.

Soll ein Bitmap, nachdem es in das Ausgaben-Fenster gesetzt wurde, noch einmal verändert werden, muss es einfach doppelgeklickt werden. Es öffnet sich wieder das Fenster Bildauswahl und Veränderungen können vorgenommen werden.



Abb. 2.36

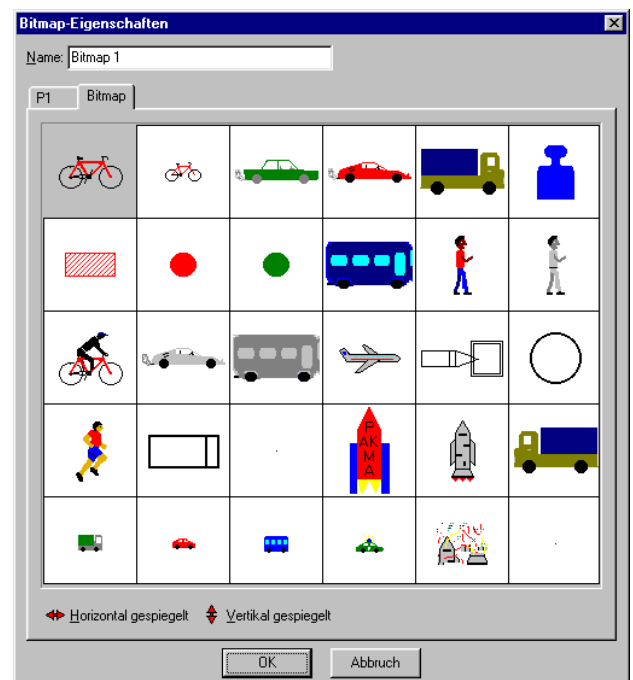


Abb. 2.37

2.6.7 Benutzerdefinierte Bitmaps

Analog zu den von PAKMA vorgegebenen Bitmaps ist es auch möglich, eigene Bitmaps in ein Ausgaben-Fenster einzufügen. Voraussetzung ist jedoch, dass diese Bitmaps nicht mehr als 256 Farben enthalten, und dass sie im Format ***.bmp** vorhanden sind. In Bezug auf die Größe des Bitmaps gibt es keinerlei Einschränkungen, im Gegensatz zu den mitgelieferten Bitmaps. Hat man diesen Unterpunkt aufgerufen, erscheint der Dialog **Datei Öffnen**, mit dessen Hilfe das gewünschte Bitmap ausgewählt werden kann.

Bevor das Bitmap in das Ausgaben-Fenster eingefügt wird hat man noch die Möglichkeit, das Aussehen des Bitmaps anzupassen. Dieser Dialog erscheint ebenfalls, wenn man auf ein bereits bestehendes benutzerdefiniertes Bitmap doppelklickt.

Neben den Möglichkeiten des **horizontalen** und **vertikalen Spiegelns**, existieren die Optionen **Zuweisen** bzw. **Exportieren**, mit denen das zu bearbeitende Bitmap durch ein anderes Bitmap ersetzt bzw. als bmp- Datei abgespeichert werden kann.

Beim **Zuweisen** wird ein Dialog **Datei Öffnen** aufgerufen, mit dem ein neues Bitmap ausgewählt werden kann

Für das **Exportieren** wird ein entsprechender Dialog **Speichern Unter** zur Verfügung gestellt.

2.6.8 Uhr

Wird dieser Unterpunkt aufgerufen, so erscheint der Cursor in Form einer Uhr. Nachdem diese im Ausgaben-Fenster platziert und auf die gewünschte Größe aufgezo- gen wurde, öffnet sich das Fenster **Uhr-Eigenschaften**.

Im Eingabefeld **gekoppelt an** kann man menügesteuert zwischen allen Ausgabegrößen auswählen. Die hier eingetragene Größe wird dann bei Ablauf des Projektes in der Uhr veranschaulicht. In das Eingabefeld **Vollkreis** ist eine Zahl einzugeben, welche den Wert des Vollkreises für die gewählte Ausgabegröße angibt. Man kann das Feld beschriften, nachdem man es mit der Maus angeklickt hat. Sinnvolle Eintragungen ergeben sich beispielsweise dann, wenn man als Ausgabegröße die Zeit t wählt und für **Vollkreis** entspricht die Zahl 60 eingibt. Die so definierte Uhr stellt dann die Zeit dar, wobei ein Umlauf dem Wert 60 entspricht.

Text 1
x
Test

Abb. 2.38

2.6.9 Text

Nachdem der Unterpunkt **Text** aufgerufen und das Textfeld durch Anklicken an die gewünschte Stelle im Ausgaben-Fenster positioniert wurde, erscheint das Fenster **Text-Eigenschaften** (s. Abb. 2.26). Hier kann in das Eingabefeld **Text** ein beliebiger Text eingegeben werden, der dann im Ausgaben-Fenster erscheint. Bei einem solchen Text kann es sich beispielsweise um eine zusätzliche Beschriftung an einem Graphen bzw. an einem anderen Animationselement (z. B. einem Vektor) handeln. Es können sowohl Buchstabenfolgen als auch Zahlen oder mathematische Operationen eingegeben werden.

Auch die **Schriftgröße** lässt sich hier einstellen, und man kann wählen, ob man den Text **fett** dargestellt haben möchte.



Abb. 2.39

2.6.10 Vielfachanimationselemente

Bei der Darstellung bestimmter physikalischer Versuche und Modelle (z. B. fortschreitende Welle mit Oszillatoren, Potential- und Stromverhalten einer Widerstandskette bei Anlegen eines Brückwiderstandes, fraunhofersche Beugung am Einzel- und Mehrfachspalt, usw.) wird eine Reihe gleichartiger Animationselemente benötigt.

Da die Größen der einzelnen Elementaroszillatoren, -auslenkungen, -wellen, usw. mathematisch geschickt mittels Indizierung beschrieben werden, ist es naheliegend, dieses Prinzip auf Animationselemente anzuwenden.

Die darzustellenden Größen werden im Kernprogramm durch Arrays beschrieben und – durch Auflistung des Array-Bezeichners im Ausgabebefehl – der Animation zugänglich gemacht. Den Komponenten der Reihenvektoren lassen sich dann durch Bestimmung eines Laufindex die zugehörigen Werte zuordnen.

2.6.10.1 Reihenvektor

Definitionen

Vektorreihe: Dies ist ein Reihenvektor, bei dem das Auswahlfeld **Vektorkette** keinen Haken hat. Die Fußpunkte der einzelnen Vektoren sind auf einer Linie äquidistant aufgereiht und im Editiermodus parallel und gleichlang und i. a. nicht hintereinander.

Vektorkette: Dies ist ein Reihenvektor, bei dem das Auswahlfeld **Vektorkette** einen Haken hat. Bei der **Vektorkette** hängen die einzelnen Pfeile hintereinander. Mit **Vektorkette** lassen sich Vektorsummen darstellen.

Erzeugen eines Reihenvektors

Nachdem im Hauptmenü **Animation** der Unterpunkt **Editieren** aktiviert wurde, ist im Menü **Animation** der Unterpunkt **Reihenvektor** verfügbar.

Nach Aufruf dieses Menüpunktes ändert sich der Mauszeiger in zwei kleine parallele Pfeile, die einen Reihenvektor symbolisieren. Zieht man nun bei gedrückter linker Maustaste den Reihenvektor größer und lässt anschließend die Maustaste los, wird ein Reihenvektor mit vier Vektoren erzeugt. Es erscheint der Dialog **Reihenvektor-Eigenschaften** (s. Abb. 2.40).

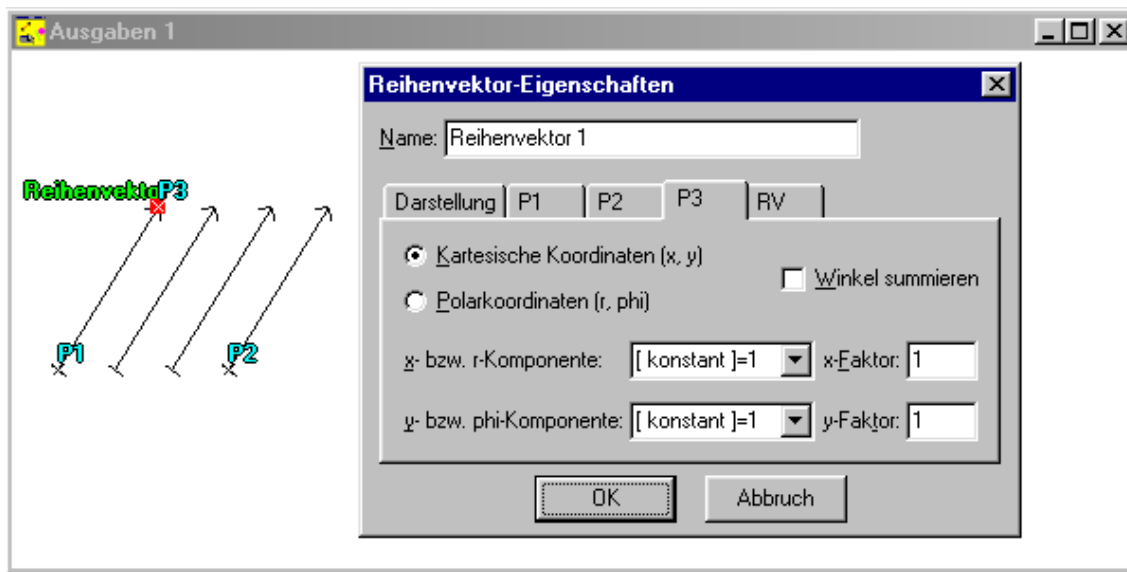


Abb. 2.40

Die Registerkarten **P1** und **P2** legen die Bewegung dieser beiden Punkte fest und bestimmen so den Anfang und das Ende der Vektorfußpunkte.

Mit Hilfe der Registerkarte **P3** lassen sich folgende Einstellungen vornehmen.

Mit den Schaltfeldern **Kartesische Koordinaten (x, y)** und **Polarkoordinaten (r, phi)** kann die Art der Koordinaten gewählt werden. Bei Einstellung auf Polarkoordinaten wird die phi-Komponente in Altgrad interpretiert.

Das Auswahlfeld **Winkel summieren** bewirkt in eingeschaltetem Zustand, dass die Richtung des einzelnen Vektors relativ zu seinem Vorgänger der Vektorreihe oder Vektorkette interpretiert wird. Dies funktioniert unabhängig von der Wahl kartesischer oder polarer Koordinaten.

In die Kombinationsfenster **x- bzw. r-Komponente** und **y- bzw. phi-Komponente** wird entweder eine bestimmte Variable oder Konstante eingetragen. Sie gilt dann für alle Vektoren der Reihe. Es kann eine einfache Variable, ein Array-Element oder eine Konstante sein, z. B. **x**, **x[3,4,5]** oder **150** oder eine Array-Komponente mit dem Laufindex **n**, z. B. **x[3,n,5]**

In diesem Fall wird dem ersten Vektor der Reihe der Wert des ersten Array-Elementes zugeordnet, mit dem diese Komponente im Kernprogramm dimensioniert wurde. (Z. B. `dim x[1..4,6..30,1..5]`, hier würde dem ersten Vektor dann `x[3,6,5]`, dem zweiten `x[3,7,5]`, dem dritten `x[3,8,5]`, usw. zugeordnet werden.)

Durch einen Klick auf den kleinen Pfeil am rechten Rand des Kombinationsfensters öffnet sich eine Aufklappliste. Der erste Eintrag dieser Liste ist die aktuelle Bezeichnung, die bei Aufruf des Eigenschaftendialogs gültig war. Darunter werden alle Variablennamen der Be-

reichsgrenzen aufgelistet. Handelt es sich um ein Array, so werden an den Variablennamen die niedrigsten Indizes des Arrays in eckigen Klammern angehängt. Steht z. B. im Kernprogramm: `dim x[1..3,4..7,3..5]`, so lautet der Eintrag `x[1,4,3]`.

Mit dem **x-Faktor** bzw. **y-Faktor** wird die horizontale bzw. vertikale Komponente der einzelnen Vektoren gestreckt bzw. gestaucht, um die graphische Darstellung der Vektorpfeile geeignet einzustellen. Die Faktoren beziehen sich hierbei auf die Darstellung im x, y – Koordinatensystem des Bildschirms.

Die errechneten Werte im Kernprogramm können sich so voll und ganz auf die physikalischen Inhalte beziehen, da die Vektoren mit diesen Faktoren skaliert werden.

Auch bei Polarkoordinaten behalten die Faktoren x und y ihre oben beschriebene Funktion. Durch Nullsetzen des x- bzw. y-Faktors werden die Vektoren auf die y- bzw. x-Komponente projiziert, ohne dass dazu im Kernprogramm die Kosinus- bzw. Sinusfunktion gebraucht würde.

Auf der Registerkarte **RV** lassen sich die Eigenschaften festlegen, die in **Abb. 2.41** zu sehen sind.

In das Kombinationsfenster **Anzahl der Elemente** wird die Anzahl der Vektoren der Reihe eingegeben. Der Vorgabewert ist 4. Es sind hierbei Werte zwischen 1 und 250 erlaubt.

Die Anzahl der Elemente lässt sich auch während der Laufzeit ändern, wenn statt einer Konstanten ein Variablenname eingegeben wird. Die gültigen Variablennamen können auch in der zugehörigen Aufklappliste ausgewählt werden.

Durch Ausschalten des Auswahlfeldes **Spitze zeichnen** kann das Zeichnen aller Vektorspitzen des Reihenvektors unterdrückt werden.

Durch Ausschalten des Auswahlfeldes **Fuß zeichnen** kann das Zeichnen aller Vektorfüße des Reihenvektors unterdrückt werden.

Im Kombinationsfenster **Breite** können Sie eine Zahl eingeben oder aus der Aufklappliste auswählen. Nach ihr richtet sich die Linienbreite der Vektorpfeile.

Mit dem Listenfenster **Farbe** kann die Strichfarbe für die Vektorpfeile eingestellt werden.

Linie von Spitze zu Spitze verbindet die Spitzen der einzelnen Pfeile mit der einzustellenden **Breite** und **Farbe**, falls das Auswahlfeld **Vektorkette** (s. u.) nicht aktiviert wird.

Linie von Fuß zu Fuß verbindet die Füße der einzelnen Pfeile mit der einzustellenden **Breite** und **Farbe**, falls das Auswahlfeld **Vektorkette** (s. u.) nicht aktiviert wird.

Vektorkette hängt den Fuß des (n+1)-ten Pfeiles an die Spitze des n-ten Pfeiles. Dadurch lässt sich eine Vektorsumme sehr leicht verwirklichen.

Wenn **Vektorkette** eingeschaltet ist, lässt sich auch der **resultierende Vektor** der Kette mit **Breite** und **Farbe** darstellen.

*Sind die Auswahlfelder **Spitze zeichnen** und **Fuß zeichnen** ausgeschaltet, lassen sich Polygone darstellen. Ist der resultierende Vektor aktiviert, sind die Polygone geschlossen.*

Wenn bei einer Vektorreihe die Linie von Spitze zu Spitze genügt (z. B. bei der Darstellung einer Welle ohne Elementaroszillatoren), so kann die Anzeige der

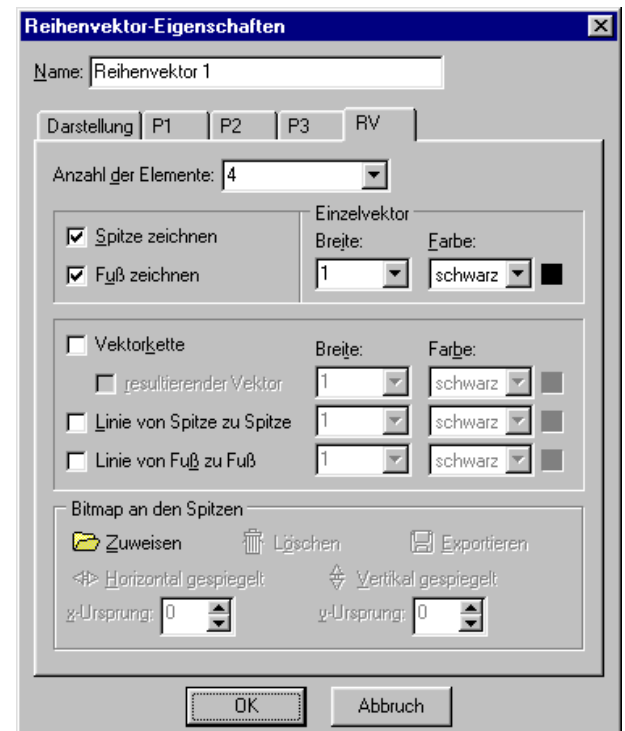


Abb. 2.41



Pfeile durch die Farbe Weiß unterdrückt werden. Dies wirkt sich auch performancesteigernd aus, da bei "weiß" die Zeichenroutinen der einzelnen Pfeile unterdrückt werden (das Zeichnen der einzelnen Pfeile kostet deutlich Zeit, vor allem bei einer Pfeilbreite größer als 1.).

Reihenvektor im Ausgaben-Fenster bearbeiten

Durch Ziehen des ersten Pfeiles wird die ganze Reihe verschoben.

Durch Ziehen des zweiten oder eines späteren Pfeiles lässt sich die Reihe dehnen. Dabei werden die Abstände zwischen den Vektoren richtig angepasst.

Wird eine Vektorkette gestreckt, so passen sich Richtung und Länge der Pfeile in natürlicher Weise an.

Sollen die Abstände der einzelnen Vektoren exakt gleich sein, dann strecken Sie die Reihe am zweiten Element. Sie vermeiden damit Kompromisse die durch unvermeidbare Rundungen entstehen, wenn die Orte der Pfeile interpoliert werden.



Alle Pfeile sind gleichlang und gleichgerichtet. Orientierung und Länge können mit den Ziehkнопfen des ersten Pfeiles verändert werden.

Durch Doppelklick auf einen Pfeil des Reihenelementes wird dessen **Eigenschaftendialog** aufgerufen.

Alternativ dazu kann mit der rechten Maustaste des Mauszeigers auf einen Pfeil des Reihenelementes geklickt werden. Daraufhin öffnet sich ein Kontextmenü, in dem der Punkt **Eigenschaften** angeklickt werden kann.

Die Ziehkнопfen der Animationselemente sind auch deren Verbindungspunkte. Die Verbindungen des Reihenvektors funktionieren wie beim Vektor.

Der Verbindungspunkt an der Spitze des ersten Vektors ist nur durch die Komponenten von **P3** bestimmt, insb. kann er nicht durch die Koordinaten eines anderen Verbindungspunktes beschrieben werden.

Bei der **Vektorkette** sitzt dieser Verbindungspunkt am Ende der Kette. Hierdurch lassen sich andere Elemente an die Vektorkette anhängen und somit auch Vektorketten aneinanderreihen.

2.6.10.2 Rechteckkette

Mit der Darstellung **Rechteckkette** kann man im PAKMA leicht eine Widerstandskette darstellen. Um jedoch die Einsatzmöglichkeiten der Rechteckkette möglichst flexibel zu halten und diese nicht auf die Darstellung einer Widerstandskette zu beschränken, wurden diesem Animationselement diverse Eigenschaften zugeordnet. Werden die Einträge der Eigenschaften extrem gewählt, so lassen sich entartete Rechteckketten erstellen (z. B. Linie mit wählbarer Breite und Farbe, Strichellinie, Sprossenleiter und gedrehtes Rechteck).

Um diese zu verdeutlichen sind in **Abb. 2.43** ausschließlich Beispiele zu unterschiedlichen Formen von Rechteckketten dargestellt.

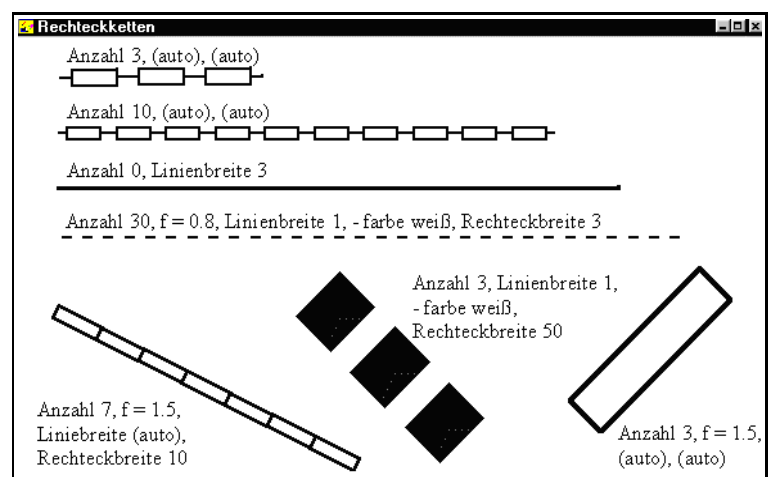


Abb. 2.42

Erzeugen einer Rechteckkette

Nachdem über das Hauptmenü **Animation** der Unterpunkt **Editieren** aktiviert wurde, ist im Menüpunkt **Animation** der Unterpunkt **Rechteckkette** verfügbar. Nach Aufruf dieses Punktes ändert sich der Mauszeiger in eine kleine Kette mit drei Rechtecken. Dies soll eine Rechteckkette symbolisieren. Es wird eine Rechteckkette mit drei (Vorgabewert) Rechtecken erzeugt.

Wie auch schon bei den anderen Animationselementen kann man jetzt die Rechteckkette auf die gewünschte Größe ziehen. Beim Loslassen der linken Maustaste erscheint der Dialog **Rechteckkette-Eigenschaften** (s. Abb. 2.43).

Werden die Einträge **[auto]** in der Dialogbox belassen, passen sich Länge, Breite und Linienstärke der Rechtecke in der Kette automatisch an die Länge der ganzen Kette an. Es entsteht dabei eine Rechteckkette, die zur Darstellung einer Kette von elektrischen Widerständen gut proportioniert ist.

Der Eigenschaftendialog Rechteckkette

Im Editierfenster **Anzahl der Rechtecke** kann man bestimmen, wie viele Rechtecke die Kette haben soll. Es sind Werte von 0 bis 250 erlaubt. Der Vorgabewert ist 3.

Um eine Linie zu erzeugen geben Sie hier '0' ein. Diese Linie kann dann in der Breite und Farbe variiert werden (s. u. **Linie – Breite/Farbe**).

Im Editierfenster **Faktor für Rechtecklänge** sind Werte von 0 bis 1,5 möglich. Mit diesem Faktor bestimmt man die relative Länge der angezeigten Rechtecke. Wird hier 0 eingetragen, so entarten alle Rechtecke der Kette zu Linien, die senkrecht zur Verbindungslinie stehen. Bleibt der Vorgabewert 1 stehen, so ergibt sich eine Kette, die z.B. als Kette von Widerständen interpretiert werden kann. Bei 1,5 stoßen die Rechtecke aneinander und eine Sprossenleiter entsteht. Wird die Länge der Kette geändert, so ändern sich auch die Längen der Rechtecke.

Im Listenfenster **Linie – Breite** können die Breite der Umrandungslinien und der Verbindungslinie manuell eingestellt werden. Der Eintrag **[auto]** stellt die Linienbreite je nach Anzahl der Rechtecke und nach Länge der ganzen Kette automatisch ein.

Im daneben liegenden Listenfenster **Farbe** kann die Linienfarbe gewählt werden.

Im Falle von **weiß** erscheinen die Rechtecke ohne Umrandung und Verbindungslinie. Hiermit lässt sich z. B. eine Strichellinie erstellen. Deren Breite wird dann mit dem nächsten Eintrag festgelegt.

Im Kombinationsfenster **Rechteck – Breite** kann die Breite der Rechtecke in Pixel eingetragen werden. Hier kann auch ein Wert eingegeben werden, der nicht in der Liste aufgeführt ist. Der Vorgabeeintrag **[auto]** wählt – abhängig von der Anzahl der Rechtecke und der Länge der Kette – automatisch eine Breite.

Im daneben liegenden Listenfenster **Farbe** wird die Füllfarbe der Rechtecke ausgewählt.

Da sich die Rechtecke der Kette parallel zu deren Verbindungslinie ausrichten, lässt sich die Rechteckkette zur Darstellung eines gedrehten Rechteckes verwenden.

Nachdem ein Animationselement in das Ausgaben-Fenster gesetzt wurde, kann man die gewünschten Veränderungen vornehmen, wie sie für die Linie beschrieben wurden. Es können gegebenenfalls die Größe verändert, das Element verschoben sowie eine neue Linien- bzw. Füllfarbe zugewiesen werden.

Außerdem können weitere Veränderungen mit den Unterpunkten des Menüpunktes **Bearbeiten** vorgenommen werden.



Abb. 2.43

Soll ein Animationselement wieder aus dem Ausgaben-Fenster entfernt werden, so muss dieses zuerst aktiviert werden. Anschließend ist der Button [], s. Abb. 1.1, bzw. der Unterpunkt **Ausschneiden** des Menüpunktes **Bearbeiten** aufzurufen.

*Man kann mehrere Elemente in einem Zug markieren, indem man bei gedrückter linker Maustaste und gleichzeitig gedrückter Umschalt-Taste (Shift) einen Rahmen im Ausgaben-Fenster aufzieht. Dann werden alle Animationselemente und Anzeigeformen der Ausgabegrößen aktiviert, die sich vollständig in dem Rahmen befinden. Ruft man nun den zuvor schon erwähnten Punkt **Ausschneiden** auf, so werden alle aktivierten Objekte im Ausgaben-Fenster gelöscht.*



Auf die gleiche Art und Weise können mehrere Objekte im Ausgaben-Fenster verschoben werden. Hierzu müssen die gewünschten Objekte zunächst markiert werden, bevor sie bei gedrückter linker Maustaste im jeweiligen Ausgaben-Fenster beliebig verschoben werden können.



Ein weiterer wichtiger Unterpunkt des Menüpunktes **Bearbeiten** ist **Neu zeichnen**. Er wird immer dann aufgerufen, wenn Veränderungen am Ausgaben-Fenster vorgenommen wurden, die dazu geführt haben, dass der Inhalt des Fensters nicht mehr exakt dargestellt wird. Wird beispielsweise ein Animationselement im Ausgaben-Fenster verschoben, so kann es vorkommen, dass es dabei eine Spur nach sich zieht. Diese Spur verschwindet wieder, wenn **Neu zeichnen** bzw. **F3** aufgerufen wird.

Sollen zu einem späteren Zeitpunkt die Eigenschaften eines Animationselementes noch einmal verändert werden, doppelklickt man auf das entsprechende Element. Allerdings muss hierzu der Editiermodus aktiviert sein. Es wird daraufhin erneut der Eigenschaften-Dialog des Elementes geöffnet und die gewünschten Veränderungen können vorgenommen werden.



2.6.11 Das Stempeln

Sinn und Zweck des Stempelns

Mit Stroboskopaufnahmen lassen sich Bewegungsabläufe gut darstellen, etwa die Aufnahme eines fliegenden Tischtennisballes bei schrägem Wurf. Die Mehrfachbelichtung in regelmäßigen Zeitintervallen erzeugt auf dem Foto eine Reihe von Tischtennisbällen, die die Form einer nach unten geöffneten Parabel hat. Dabei ist der Abstand eines Balles zu seinem Nachbarn ein Maß für seine Geschwindigkeit an diesem Ort.

Um diese Möglichkeit auch in PAKMA anzubieten, wurde das Stempeln implementiert.

Man kann diese Funktion ein- und ausschalten sowie die Anzahl der Ausgabebefehle einstellen - die Schrittweite, die intern aufgerufen werden müssen bis wieder gestempelt wird. Alle Animationselemente, deren Attribut auf **Stempeln** gesetzt wurde, nehmen dann an diesem Prozess teil.

Abb. 2.44 wurde mit PAKMA erzeugt und zeigt einen schrägen Wurf unter Berücksichtigung des Luftwiderstandes. Bewegt und dann gestempelt wird das Animationselement Kreis dessen Ortskoordinaten nicht fixiert, sondern eine Funktion der Zeit sind.

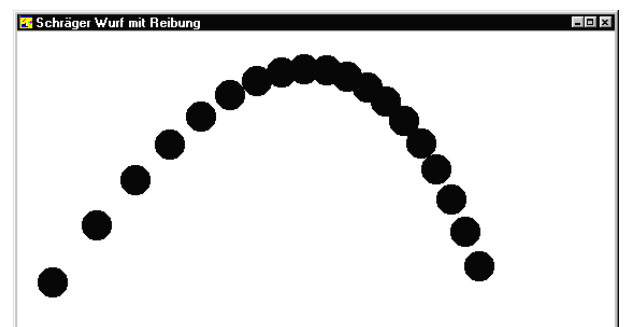


Abb. 2.44

Anleitung für das Stempeln

Mit **Animation → Stempeln...** im Hauptmenü wird der Dialog **Stempeln** geöffnet:

Mit dem Auswahlfeld **Stempeln** schaltet man die Stempelfunktion zentral ein oder aus.

Die Anzahl der Aufrufe des Ausgabebefehls, die nach jedem Stempeln vergehen, wird im Kombinationsfenster **Schrittweite** eingestellt. Steht z. B. in Schrittweite die Zahl 4, so wird beim ersten Ausgabebefehl gestempelt und danach dreimal nicht. Dieser Vorgang wiederholt sich zyklisch.

Die Funktion Stempeln wird nur auf die jeweils ausgewählten Elemente angewendet.



Um ein Animationselement für das Stempeln auszuwählen, muss zuerst dessen Attribut **Stempeln** gesetzt werden. Dazu wird der Mauszeiger auf das Element positioniert und die rechte Maustaste geklickt oder die Zuordnung wird über die Elemente-Übersicht, s. Abb. 2.46 vorgenommen.

Es erscheint eines der beiden folgenden Popup-Menüs:

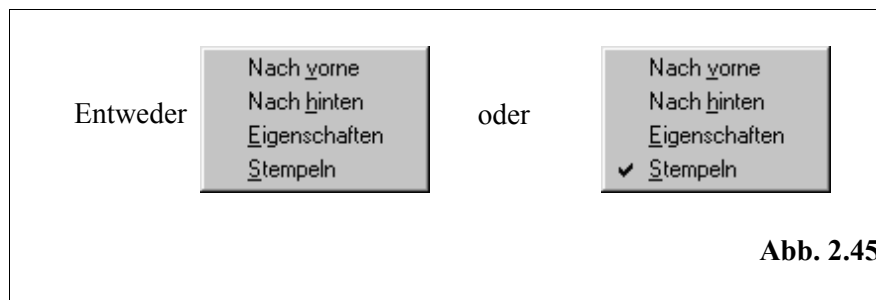


Abb. 2.45

Mit einem Klick auf **Stempeln** ändern sich dessen Attribut. Wird ein Haken vor **Stempeln** angezeigt, ist es bereits gesetzt, und das Animationselement wird bei jeder Stempelaktion berücksichtigt.

Da die Einstellungen der einzelnen Unterpunkte ähnlich vorzunehmen sind, soll im Folgenden nur an einigen Animationselementen die prinzipielle Vorgehensweise exemplarische aufgezeigt werden.

2.6.12 Überblick über verwendete Elemente

Wenn man den Unterpunkt **Elemente-Übersicht** im Menüpunkt **Animation** anklickt oder die Taste **F8** drückt, erscheint ein Fenster, in dem alle verwendeten Animationselemente übersichtlich nach **Name**, **Typ** oder **Position** aufgelistet werden. Durch Anklicken eines Feldes in dieser Übersicht werden die Markierungspunkte des Elementes und sein Name auf dem Bildschirm angezeigt. Daher ist es wichtig, den Animationselementen markante Namen zu geben, so dass sie aus der Übersicht leicht wieder erkannt werden. Durch Doppelklicken des Feldes kann der zugehörige **Eigenschaften**-Dialog sofort aufgerufen werden, was auch geschieht, wenn man das zugehörige Symbol in der untersten Leiste anklickt. Dort läßt sich außerdem einstellen, ob das jeweils markierte Element **gestempelt** werden soll und wie weit es im **Hintergrund** bzw. **Vordergrund** liegen soll.

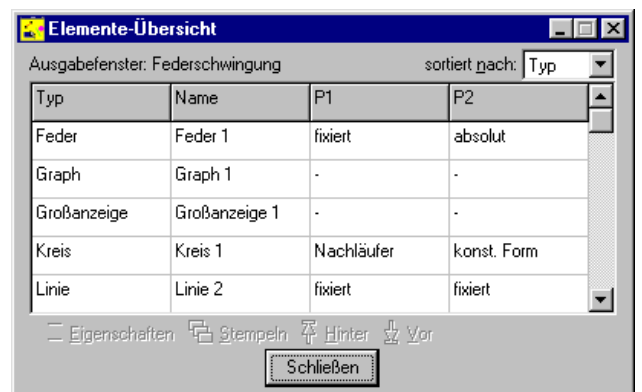


Abb. 2.46

2.6.13 Überblick über die eingetragenen Beziehungen

Mit dem Unterpunkt **Beziehungen zeigen** des Menüpunktes **Verbinden** werden alle Markierungspunkte der Animationselemente sowie Verbindungslinien zwischen den Masterpunkten und Nachläufern eingezeichnet. Man kann damit kontrollieren, welche Verbindungen den Markierungspunkten zugeordnet wurden bzw. bei welchen Punkten es sich um Masterpunkte oder Nachläufer handelt.

Einen **Masterpunkt** erkennt man daran, dass er durch eine Linie mit einem Nachläufer verbunden ist.

Die eingezeichneten Markierungspunkte und Verbindungslinien werden wieder gelöscht, wenn man den Bildschirm an einer beliebigen Stelle anklickt.

*Sollen Veränderungen an einem Animationselement vorgenommen werden, nachdem ein Projekt bereits einen Durchlauf absolviert hat, müssen zunächst die Unterpunkte **Editieren** der Menüpunkte **Ausgaben** und **Animation** aufgerufen werden. Erst dann kann man die gewünschten Veränderungen durchführen.*

Um zu veranschaulichen, wie man beim Erstellen einer dynamischen Repräsentation vorzugehen hat, wird im Folgenden das bereits bekannte Projekt **Federschwingung.prj** zu dem physikalischen Vorgang der „Schwingung einer Masse an einer Feder“ Schritt für Schritt erstellt.

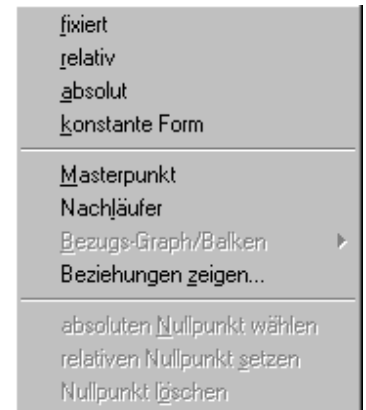


Abb. 2.47



2.7 Erstellen des Projektes *Federschwingung*

2.7.1 physikalisches Modell erstellen

Bei unserem Modell soll eine Masse m an einer Feder (Federkonstante k) schwingen (s. **Abb. 2.48**). Es soll untersucht werden, wie sich die Auslenkung y , die Geschwindigkeit v und die Beschleunigung a der Masse in Abhängigkeit von der Zeit t verhalten. Dabei soll die Masse zu Beginn (Zeitpunkt $t=0$) eine Anfangsauslenkung y_0 besitzen, von der aus die Schwingung beginnt. Die Berechnung der Größen y , v und a soll iterativ über die Zeit erfolgen. Die Schrittweite zwischen den Berechnungen von y , v und a sei hierbei dt (Zeitschritt).

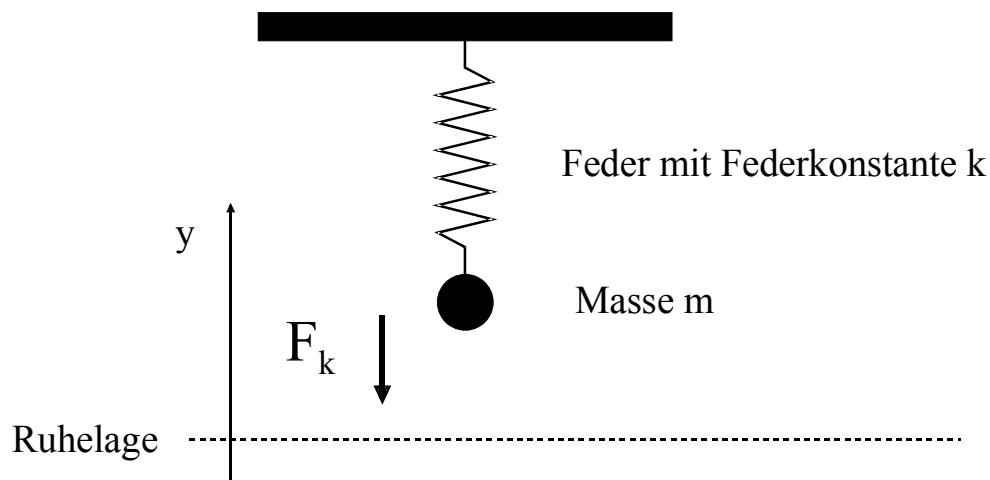


Abb. 2.48

Der physikalische Sachverhalt lässt sich durch einen Kraftansatz beschreiben. Die rücktreibende Kraft, die die Masse m beschleunigt, ist hier die Federkraft F_k :

$$F_k = m \cdot a = -k \cdot y$$

Zunächst werden die Startwerte und Konstanten festgelegt:

```
m = 1           // Masse
k = 1           // Federkonstante
dt = 0.05       // Zeitschritt für die Iteration
t = 0           // Zeitpunkt des Starts
y = 2           // Anfangsauslenkung
v = 0           // Anfangsgeschwindigkeit
```

Mit Hilfe des Hookschen Gesetzes und der Grundgleichung der Mechanik können nun a , v und y iterativ berechnet werden:

$$F = -k \cdot y$$

$$a = F/m$$

$$v = v + dv \text{ und } dv = a \cdot dt \quad \Rightarrow \quad v = v + a \cdot dt$$

$$y = y + dy \text{ und } dy = v \cdot dt \quad \Rightarrow \quad y = y + v \cdot dt$$

2.7.2 Kernprogramm schreiben

Bereits im ersten Schritt wurde festgelegt, dass die iterative Berechnung der Ausgabegrößen in unserem Beispiel in Abhängigkeit von der Zeit t erfolgen soll.

D. h., die **REPEAT-UNTIL-Schleife** innerhalb des Kernprogramms wird über die Zeit t laufen. Die maximale Ablaufzeit mt und der Zeitschritt dt werden als Konstanten (neben anderen) zu Beginn des Kernprogramms definiert.

Auf welche Art und Weise man die Ausgabegrößen berechnen kann, wurde ebenfalls bereits erläutert.

```
t:=0; y:=1; v:=0; //Anfangswerte für t, y, v
m:=1; k:=1; //Masse m und Federkonstante D
dt:=0.05;mt:=10; //Zeitintervall dt und Ablaufzeit mt
REPEAT //Schleifenbeginn
    F:= -k*y; //Berechnung von a, y, v
    a:= F/m;
    y:= y+v*dt;
    v:= v+a*dt;
    t:= t+dt; // Erhöhung von t um dt
    ausgabe (t,y,v,a );// Ausgabeanweisung für t, y, v, a
UNTIL t>mt; // Abbruchbedingung der Schleife
```

Da in unserem Beispiel das Verhalten der Größen y , v und a in Abhängigkeit von t untersucht und später in Ausgaben-Fenstern dargestellt werden soll, müssen als Argumente in der Ausgabeanweisung die Größen y , v , a und t aufgeführt werden.

Man hat nun die Möglichkeit dieses Kernprogramm in das Fenster **Editor** einzugeben. Wenn dieses Fenster geöffnet und aktiv ist, erscheint ein blinkender Cursor in der ersten Zeile. Die Eingabe erfolgt dabei über die Tastatur des Rechners. Alternativ kann man aber auch das Kernprogramm, wenn man es in einem Textprogramm erstellt hat, direkt in den **Editor** übernehmen, wenn man den Programmtext markiert und in die Zwischenablage kopiert und anschließend in PAKMA den Unterpunkt **externes Kernprogramm** des Menüpunktes **Bearbeiten** aufruft.

Wenn das fertige Projekt komfortabler z. B. mit Hilfe von Schiebern bedient werden soll, müssen noch einige Ergänzungen in der Programmstruktur vorgenommen werden. Dies wird im abschließenden Abschnitt 2.8.6 ausgeführt.

Ein kleiner Hinweis am Rande: Es geht häufig schneller, das Kernprogramm eines bestehenden Projektes abzuändern und den eigenen aktuellen Bedürfnissen anzupassen, als es völlig neu zu schreiben.



2.7.3 Startwerte in Menü eintragen

Die Anfangswerte und Konstanten sollen nun im Startwerte-Fenster definiert werden. Dieser Arbeitsschritt ist nur dann nötig, wenn man später die Startwerte verändern möchte. Man könnte sie zwar auch im Kernprogramm ändern, was aber sehr viel unübersichtlicher wäre und im Sofortlaufmodus gar

Startwerte (editierbar)				
Variable	Bezeichnung	Vorgabe	Letzter W.	Neuer W.
dt	Zeitintervall	0.05	0.05	0.05
mt	Meßzeit	10	10	10
m	Masse in kg	10	10	10
k	Federkonst N/m	10	10	10
v	Anfangsgeschwindigkeit	0	0	0

Abb. 2.49

nicht möglich ist. Schließlich ist es die einzige Lösung, wenn später ohne Compiler gearbeitet werden soll.

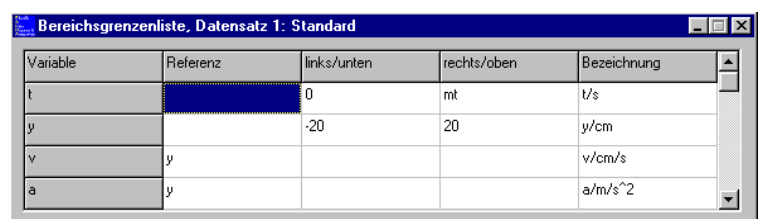
Das Startwerte-Fenster kann dann folgendermaßen aussehen (s. **Abb. 2.49**):

Es müssen die jeweiligen Wertzuweisungen im Kernprogramm entfernt werden, da die Werte aus dem Startwertemenü beim Compilieren an den Anfang des Kernprogramms gestellt werden und somit durch Zuordnungen im Kernprogramm selbst überschrieben werden.

Zuweisungen, die Variablen auf Null setzen, können i. Allg. entfallen, da nichtinitialisierte Variablen von PAKMA automatisch den Wert Null zugewiesen bekommen.

2.7.4 Bereichsgrenzen festlegen

Nun sollen die Ausgabegrößen y und v in Abhängigkeit von t graphisch dargestellt werden. Damit müssen für y , v und t Bereichsgrenzen festgelegt werden. Für die Ausgabegröße a wird keine Bereichsgrenze eingetragen. Das Programm wählt also automatisch -1 und $+1$ als Bereichsgrenzen für a .



Variable	Referenz	links/unten	rechts/oben	Bezeichnung
t		0	mt	t/s
y		-20	20	y/cm
v	y			v/cm/s
a	y			a/m/s ²

Abb. 2.50

Für die Größen y , v und t schätzt man zunächst maximale und minimale Werte, die während des Ablaufs des Projektes angenommen werden und legt dann dementsprechend sinnvolle Bereichsgrenzen fest.

für t

Die Zeit t läuft bei diesem Projekt von 0 bis zur maximalen Zeit mt , wobei mt in den Startwerten als Konstante ($mt:=10$) definiert wurde. Die Achse für t in der graphischen Darstellung soll so skaliert werden, dass der gesamte Versuchsablauf im Diagramm erfasst wird. Da mt eine Konstante ist, bietet es sich an, diese als obere Bereichsgrenze zu verwenden. Das hat den Vorteil, dass für verschiedene Versuchsabläufe, bei denen mt verändert werden soll, keine neuen Bereichsgrenzen für t eingegeben werden müssen. Als Bereichsgrenzen für t wählt man nun Null und mt . Den Zahlenwert Null trägt man in die Spalte **links/unten**, die Variable mt in die Spalte **rechts/oben** ins Fenster **Bereichsgrenzenliste** ein.

für y

Für die Auslenkung y wurde als minimaler Wert der Anfangswert -20 cm festgelegt. Da es sich bei unserem Projekt um eine ungedämpfte Schwingung handelt, ergibt sich für die Auslenkung y der maximale Wert $+20$ cm. Als Bereichsgrenzen können wir hier den minimalen und maximalen Wert wählen. Man trägt also $+20$ und -20 in die entsprechenden Spalten in der Tabelle ein.

für v

Da für die Masse m und die Federkonstante k jeweils der Wert 1 gewählt wurde, liegen die Werte für v , ebenso wie die Werte für y , zwischen -2 und $+2$.

Dies lässt sich mit Hilfe der Lösung der Schwingungsdifferentialgleichung

$$v_{\max} = \omega y_{\max} \text{ erklären, wobei hier } \omega = \sqrt{\frac{k}{m}} \text{ die Kreisfrequenz darstellt.}$$

Man kann also die Zahlenwerte -20 und $+20$ als Bereichsgrenzen in die entsprechenden Spalten eintragen.

Diese Schritte müssen jedoch nicht notwendigerweise durchgeführt werden, es genügt auch, die Bereichsgrenzen für v einfach abzuschätzen und sie nach Ablauf des Projektes, falls erforderlich, zu ändern oder die Autoskalierung zu nutzen, s. Kap. 2.4.4.



Da die Bereichsgrenzen von v und y übereinstimmen, und sie für y bereits eingegeben wurden, bietet es sich an, für v in die Spalte **Referenz** die Variable y einzutragen. Damit erreicht man, dass v ebenso wie y skaliert wird, ohne dass konkrete Bereichsgrenzen für v eingegeben werden müssen.

Nachdem man die Bereichsgrenzen festgelegt und eingegeben hat, werden die Werte nach dem Schließen des Fensters übernommen.

Festlegen der Bereichsgrenzen durch Autoskalierung

Damit man die Bereichsgrenzen von Ausgabegrößen automatisch vom Programm festlegen lassen kann, muss das Kernprogramm zuvor mindestens einmal abgelaufen sein. Dann stehen dem Rechner Werte für die Ausgabegrößen zur Verfügung, aus denen die Bereichsgrenzen bestimmt werden können. Es wird daher an dieser Stelle darauf verzichtet, s. Kap. 2.4.4.

Alternative Datensätze von Bereichsgrenzen

Möchte man beim Federschwinger aufzeigen, wie sich z. B. die Größe der Anfangsauslenkung y auf die Bewegung auswirkt, so kann man dazu zusätzliche Projektdurchläufe mit sehr kleiner Anfangsauslenkung y durchführen. Es bietet sich an, zuvor für diesen Fall einen weiteren Datensatz von Bereichsgrenzen festzulegen. Zwischen den Durchläufen wechselt man dann im Fenster **Startwerte** den Wert für die Anfangsauslenkung und im Fenster **Bereichsgrenzenliste** den Datensatz für die Bereichsgrenzen. Solche unterschiedlichen Datensätze lohnen sich insbesondere dann, wenn mehrere Graphen dargestellt werden sollen und mehrfach zwischen den unterschiedlichen Durchläufen hin- und hergesprungen wird. Wie man hierbei vorgeht, wurde bereits im vorhergehenden Abschnitt ausführlich behandelt und wird hier nicht weiter ausgeführt.

2.7.5 Ausgaben-Fenster gestalten

Obwohl schon einige Arbeitsschritte absolviert wurden, ist das Ausgaben-Fenster immer noch leer. Dies soll sich jedoch im Folgenden ändern. Zunächst sollte man sich überlegen, welche Animationselemente im Ausgaben-Fenster erscheinen sollen.

In unserem Fall soll die Schwingung eines Federpendels dargestellt werden. Hierzu werden zunächst die **Animationselemente** und **Ausgaben** im Ausgaben-Fenster platziert, wie dies für die Feder in Abb. 2.51 zu entnehmen ist. Im dargestellten Ausgaben-Fenster befinden sich eine **Feder**, ein **Kreis**, ein **Rechteck** und eine **Großanzeige**.



Abb. 2.51

Für sämtliche Animationselemente sind beim Erstellen Einstellungen in den jeweiligen Eigenschaften-Dialogen vorzunehmen, so wie dies den jeweiligen Abbildungen zu entnehmen ist.



Abb. 2.53

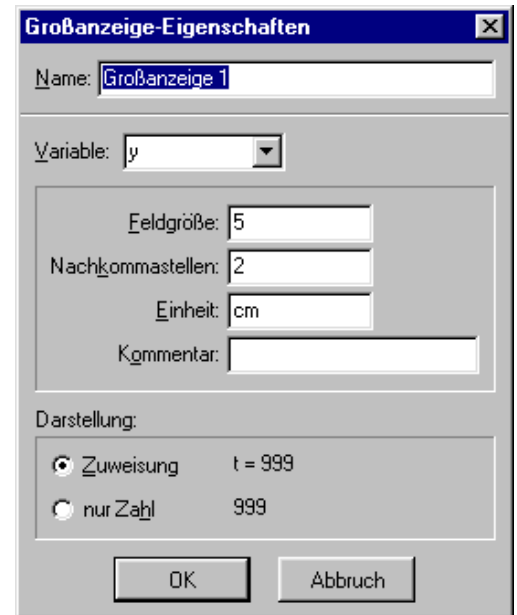


Abb. 2.52

Als nächstes wird die **Graphendarstellung** eingefügt, doch bevor diese im Ausgaben-Fenster erscheint, sind die zugehörigen Dialoge **Graphenliste** und **Eigenschaften Graphendarstellung** so einzustellen, wie man dies Abb. 2.53 und Abb. 2.54 entnehmen kann.



Abb. 2.54

Sind alle Einstellungen beendet, sollte das Ausgaben-Fenster von PAKMA aussehen, wie dies Abb. 2.55 zu entnehmen ist.

Die einzelnen Animationselemente sind mit Ausgabegrößen verbunden. Dies bedeutet, dass der Mittelpunkt des Kreises **absolut** mit der Ausgabegröße **y** verbunden wird, wohingegen der zweite Punkte des Kreises eine **konstante Form** bekommt. Der Mittelpunkt wird außerdem als **Masterpunkt** für das untere Ende der Feder definiert, das zum **Nachläufer** gemacht wird.

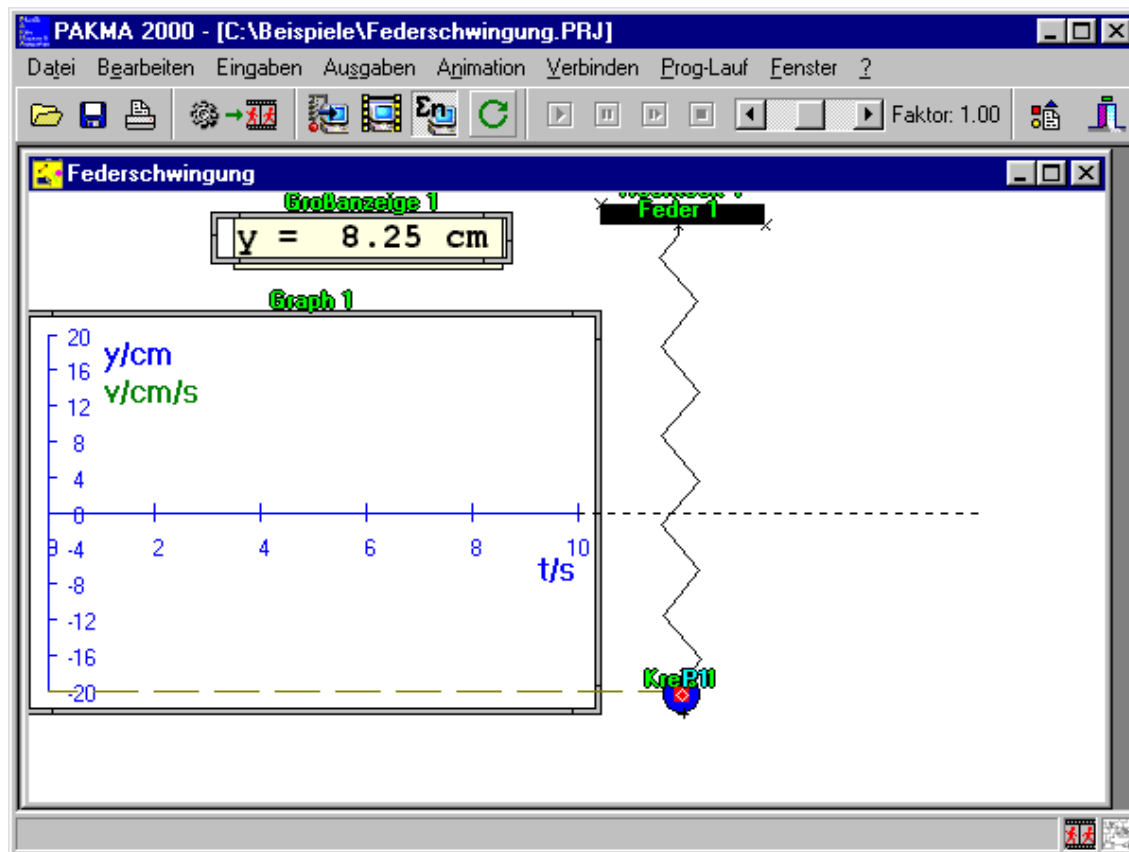


Abb. 2.55

Sämtliche Beziehungen können mit Hilfe des Unterpunktes **Beziehungen zeigen...** des Menüpunktes **Verbinden** im Ausgaben-Fenster angezeigt werden.

Damit bei einem evtl. Fehler in den Einstellungen keine Daten verloren gehen, sollte das Projekt vor einem Probelauf abgespeichert werden



Als nächstes werden alle restlichen Animationselemente hinzugefügt. Dies wären im Einzelnen eine **Linie** vom Mittelpunkt des bereits vorhandenen Kreises zum (t,y) Punkt der y(t)-Graphendarstellung, diverse **Vektoren** und **Texte**. Wie man hierbei vorgehen muss, wird im Folgenden ausführlich erläutert.

2.7.5.1 Texte

Am einfachsten lassen sich die Texte einfügen. Wenn sich diese an der entsprechenden Position im Ausgaben-Fenster befinden, wird zunächst der Mittelpunkt des Kreises zum **Masterpunkt** gemacht und sämtliche Punkte der Texte werden seine **Nachläufer**. Hierdurch wird erreicht, dass sich die Texte immer auf gleicher Höhe zum Mittelpunkt des Kreises befinden.

2.7.5.2 Linie

Das rechte Ende der eingezeichneten **Linie** wird ebenfalls **Nachläufer** des Kreis-Mittelpunktes. Das linke Ende soll nun **absolut** mit den Ausgabegrößen **t** und **y** verbunden werden. Nach Aktivierung des Punktes **absolut** des Menüpunktes **Verbinden** der Dialog, den Abb. 2.57 zeigt. Sämtliche Einstellungen sind so vorzunehmen, wie dies aus der



Abb. 2.56

Abbildung ersichtlich ist. Da sich der Endpunkt wie der Punkt im $t \rightarrow y$ -Graph bewegen soll, er-

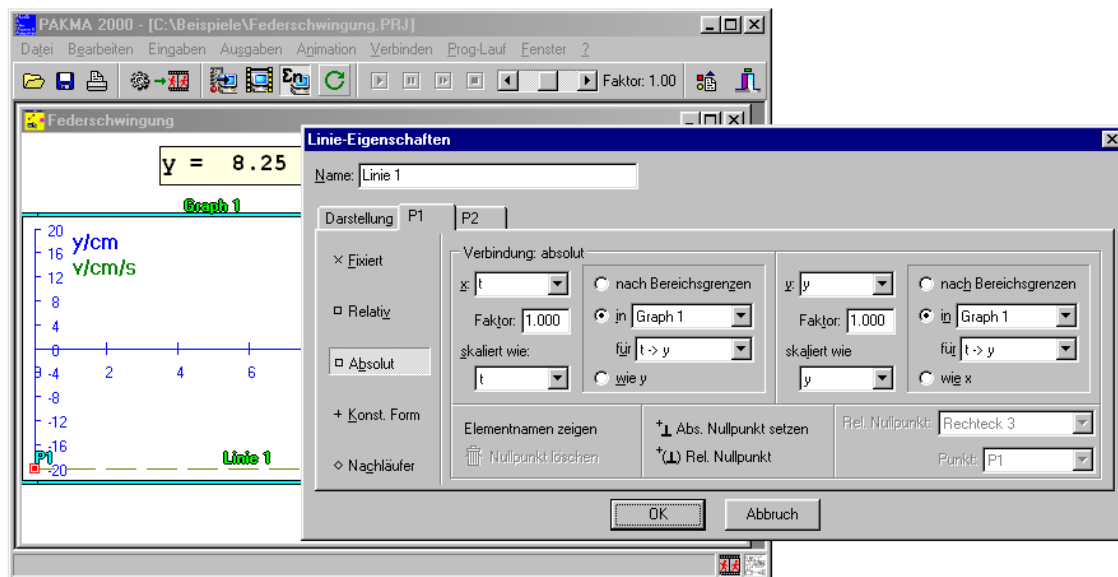


Abb. 2.57

folgt die Skalierung hier nicht nach Bereichsgrenzen sondern entsprechend dem zugehörigen Graph 1. Wichtig ist, dass für x bzw. y im Feld: 'skaliert wie' die gleiche Variable stehen muss wie die 1-te bzw. 2-te Variable im Feld 'für', das den zugehörigen Graphen angibt.

2.7.5.3 Vektoren

Beim Erzeugen des Geschwindigkeitsvektors wird ein Dialog aufgerufen, bevor man den Vektor im Ausgaben-Fenster platzieren kann. Hier wird der **y-Komponente** die Ausgabegröße v zugewiesen. Die **x-Komponente** soll den Wert 0 erhalten. Am einfachsten ist das dadurch zu erreichen, dass im Variablenfeld [Versatz] ausgewählt und im darunterstehenden Feld: Pixel der Wert 0 eingetragen wird, d. h. Die x -Komponente wird 0 Pixel groß gewählt bekommt zwar die Ausgabegröße t zugeordnet, der **Faktor** wird hier aber auf den Wert **Null** eingestellt, da sich der Vektor nur in vertikaler Richtung bewegen soll. Nach Drücken des Buttons **OK** befindet man sich wieder im Ausgaben-Fenster.

Die **Fußpunkte** der Vektoren werden zu **Nachläufern** des Mittelpunktes des Kreises, der zuvor wieder zum **Masterpunkt** gemacht werden muss.

In **Abb. 2.58** werden alle Beziehungen noch einmal angezeigt. Man erhält diese Ansicht, wenn im Menüpunkt **Verbinden** der Unterpunkt **Beziehungen zeigen** gedrückt wird.

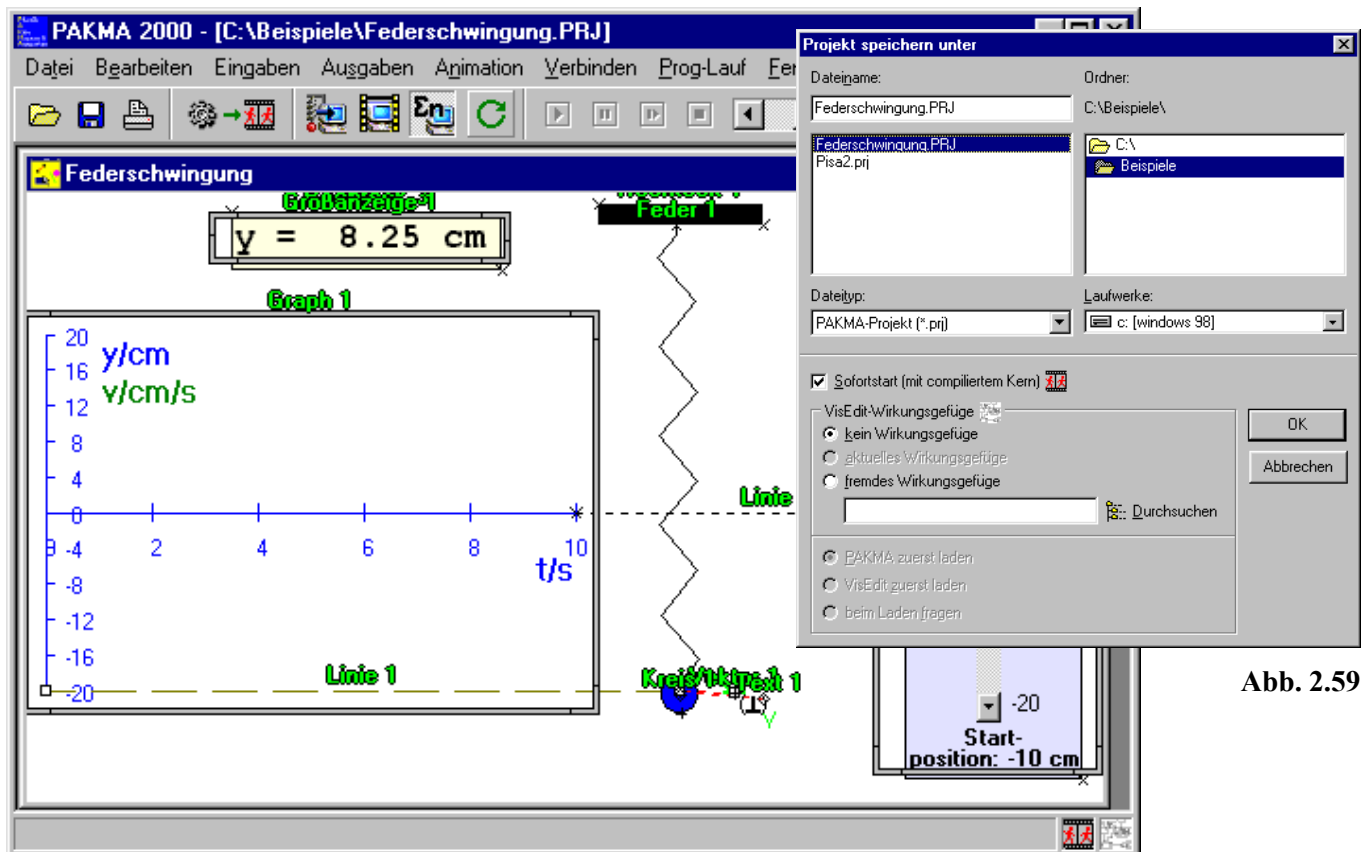



Abb. 2.59

Abb. 2.58

Bevor man einen Programmlauf durchführt, sollte das fertige Projekt unbedingt abgespeichert werden. Hierbei sollte man beachten, dass der richtige Modus eingestellt ist und dass das Projekt mit bereits compilierten Kern abgespeichert wird, so dass es beim nächsten Aufruf sofort startbereit ist. (s. Abb. 2.59)

Nun kann mit dem fertigen Projekt ein Programmlauf durchgeführt werden, indem man den Unterpunkt **Start als Modell** aufruft, die Taste **F11** drückt oder am einfachsten den Startbutton  betätigt.

Wenn alles richtig gemacht wurde, sollte am Ende der Bildschirm so aussehen, wie dies in **Abb. 2.60** gezeigt wird.

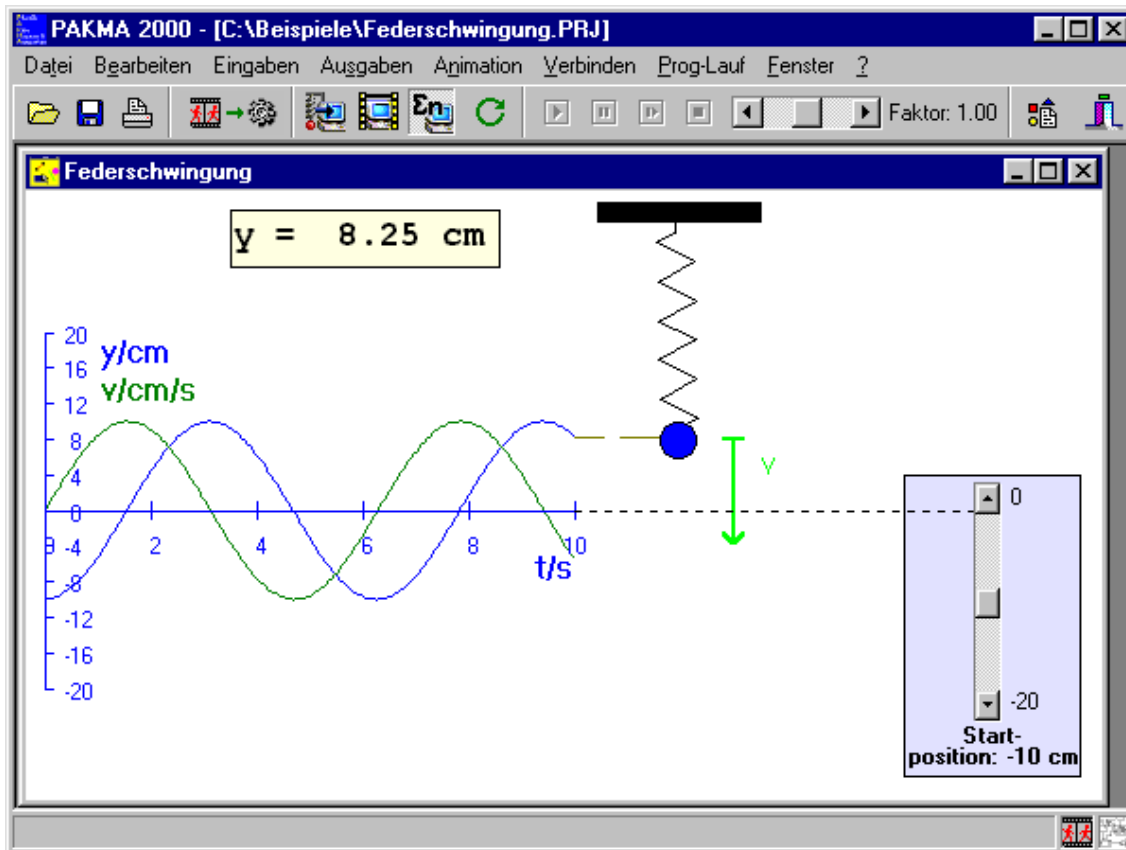





Abb. 2.60

2.7.6 Bedienungsfreundliche Ergänzungen im Kernprogramm

Damit ist das Projekt bereits voll einsatzfähig, jedoch lässt sich die Bedienung des Programms noch komfortabler gestalten. Deshalb kann es ratsam sein, einige Änderungen in das Kernprogramm einzubauen, die recht einfach zu realisieren sind, da Sie immer die gleich formale Struktur aufweisen.

Der erste Schritt besteht darin die konstanten Anfangswerte in die PAKMA-Startwerte zu übernehmen (siehe Abschnitt 2.1.4 bzw. 2.8.3), um diese auch im Sofortstartmodus variieren zu können. Danach sollten Sie die Möglichkeit der komfortableren Bedienung eines Projektes über die Buttonleiste im PAKMA-Ausgabefenster ausnutzen und die **Ablauf-Taste**  ins Kernprogramm mit einbeziehen, die intern über die Bezeichnung **start__** abgefragt wird. Hierzu müssen Sie den Zustand der **Ablauf-Taste**  über die Entscheidung, ob **start__** = 0 oder = 1 ist abfragen. Dabei ist die **Ablauf-Taste**  als **Taster** definiert, der nach einmaligem Anklicken in seine Ruheposition zurückkehrt. Daher ist es nötig, eine Variable (im Beispiel: **anfang=1**) auf einen bestimmten Wert zu setzen, um die Berechnung zu starten und bis zum vorgegebenen Wert durchlaufen zu lassen. Somit könnte Ihre erste „Optimierung“ des Kernprogramms folgendermaßen aussehen:

```

C:\PAKMA2000\temp\temp.pkp [editierbar]
Kernprogramm:
//      Simulation einer Federschwingung
;
t:=0;           ;//Starten zum Zeitpunkt t=0
y:=ya;         ;//Anfangsauslenkung

repeat         ;//Schleifenbeginn

  //Beginn der physikalischen Berechnungen
  ;
  F:=-k*y;     ;//a,F Berechnung
  a:=F/m;

  if start =1 then anfang:=1 ;//Abfrage der Ablauf-Taste
  if anfang=1 then           ;//Berechnung erfolgt erst nach Betätigen der Ablauf-Taste
  begin
    v:=v+a*dt;               ;//Berechnung der Geschwindigkeit
    y:=y+v*dt;               ;//  -" -      des Ortes
    t:=t+dt;
  end;
  ausgabe (t, y, v, a);      ;//Angabe der Ausgabevariablen
until t>mt;


```



 **Zusätzliche
Ergänzung**

Abb. 2.61

Alle beschriebenen Kernprogramme befinden sich abgespeichert (als Textdatei) auf der CD, so dass diese kopiert, in ein geöffnetes PAKMA-Editorfenster eingefügt und ausgestaltet werden können. Ebenso ist die „grobe“ Schleifenstruktur, sozusagen als „Rohprogramm“ mitabgespeichert.

 **Hinweis**

Mit Hilfe dieser kleinen Erweiterung ist die Bedienung schon wesentlich angenehmer, da auch im **Sofortstartmodus** zunächst das statische Bild der Simulation, das den gewählten Startwerten entspricht, mit den Schülern betrachtet werden kann und die Schüler Vorhersagen zum Ablauf der nun folgenden Simulation machen können. Anschließend kann durch Anklicken der **Ablauf-Taste**  die Animation gestartet werden.

Im nächsten Schritt soll die für häufige Variationen doch etwas umständliche Bedienung über die **Startwerte** vereinfacht werden. Dazu dienen die in Kapitel 2.5 beschriebenen **Schieber**. In unserem Beispiel soll die Anfangsauslenkung der Feder über einen Schieber *vor* dem Ablauf der Simulation einstellbar sein, um deren Einfluss zu verdeutlichen. Im Kernprogramm muss folglich die Anfangsauslenkung ya durch die Schiebereinstellung definiert werden. Der zugehörige Befehl lautet: **y:=schieber(ya)**. Da der Schieber nur *vor* Betätigen der **Ablauf-Taste**  einstellbar sein sollte (was auch dem entsprechenden physikalischen Experiment entspricht), muss ihm ebenfalls, wie bei der Berechnung nach Anklicken der **Ablauf-Taste** , eine **if**-Bedingung (im Beispiel: **anfang=0**) zugewiesen werden. Damit ergibt sich folgendes Kernprogramm:


```

C:\PAKMA2000\temp\temp.pkp (editierbar)
Kernprogramm:
//      Simulation einer Federschwingung
;
t:=0;           ;//Starten zum Zeitpunkt t=0
anfang:=0;      ;//Variable zur Unterscheidung, ob Ablauf-Taste betätigt wurde

repeat          ;//Schleifenbeginn

  //Beginn der physikalischen Berechnungen
  ;
  F:=-k*y;      ;//a,F Berechnung
  a:=F/m;

  if start__=1 then anfang:=1 ;//Abfrage der Ablauf-Taste

  if anfang=0 then ;//VOR Betätigung der Ablauf-Taste
  begin
    y:=schieber (ya); ;//Schieber zur Einstellung der Anfangsauslenkung
  end

  if anfang=1 then ;//NACH Betätigung der Ablauf-Taste
  begin
    v:=v+a*dt;      ;//Berechnung der Geschwindigkeit
    y:=y+v*dt;      ;//  --  des Ortes
    t:=t+dt;
  end;
  ausgabe (t, y, v, a); ;//Angabe der Ausgabevariablen

until t>mt;

```



**Zusätzliche
Ergänzung**

Abb. 2.62

Nähere Einzelheiten zur Gestaltung des oder der Ausgabefenster mit Schiebern entnehmen Sie bitte Kapitel 2.8.5 bzw. 2.5!



Hinweis

Natürlich könnten Sie noch andere Anfangswerte über Schieber einlesen, indem Sie in der zu **anfang=0** gehörige **if**-Schleife weitere Schieber definieren. Ebenso wäre es möglich Schieberstellungen während des gesamten Projektablaufes abzufragen (was allerdings in dem Ihnen vorliegenden Beispiel keinen Sinn macht). Dazu müssten Sie den Schieber außerhalb der **if**-Schleifen, also innerhalb der **repeat**-Schleife setzen. Natürlich könnten Sie den Schieber auch nur während des Ablaufs auslesen, wenn Sie diesen in die **if**-Schleifen zu **anfang=1** positionieren.

Im letzten Schritt der „Verbesserung“ eines Kernprogramms sollen noch zwei Möglichkeiten beschrieben werden, mehrere Graphen eines simulierten physikalischen Vorgangs in *ein* Diagramm einzuzeichnen.

Um dies ohne den Umweg über die **Graphik-Datenablage** zu erreichen, muss im bisherige Kernprogramm nur eine Abbruchbedingung für die Dauer der Berechnung eingegeben werden, nach der das Modell in den Anfangszustand zurückkehrt und dann der Ablauf wieder neu ausgelöst werden muss. Für den neuen Ablauf müssen vorher Anfangswerte und/oder Schieber-einstellungen neu zugewiesen werden. Daher sind die **Startwerte** nach jedem Durchlauf neu einzulesen und die Graphen neu einzuzeichnen (Befehl: **neu_graf**). Umgesetzt ergibt dies folgendes Kernprogramm:

```

C:\PAKMA2000\temp\temp.pkp (editierbar)
Kernprogramm:
//      Simulation einer Federschwingung
;
anfang:=0;                ;//Variable zur Unterscheidung, ob Ablauf-Taste betätigt wurde
repeat                    ;//Schleifenbeginn

  //Beginn der physikalischen Berechnungen
  ;
  F:=-k*y;                ;//a,F Berechnung
  a:=F/m;

  if start__=1 then anfang:=1; ;//Abfrage der Ablauf-Taste

  if anfang=0 then        ;//VOR Betätigung der Ablauf-Taste
  begin
    t:=0;                ;//Starten zum Zeitpunkt t=0
    startwerte;          ;//erneutes Einlesen der Startwerte
    y:=schieber (ya);     ;//Schieber zur Einstellung der Startposition
  end;

  if anfang=1 then        ;//NACH Betätigung der Ablauf-Taste
  begin
    v:=v+a*dt;           ;//Berechnung der Geschwindigkeit
    y:=y+v*dt;           ;//    -"-    des Ortes
    t:=t+dt;
  end;
  ausgabe (t, y, v, a);   ;//Angabe der Ausgabevariablen



  if (t>mt) then          ;//WENN ein Durchlauf der Simulation (mt) erfolgt ist, DANN
  begin
    neu_graf;            ;//Graphen sollen neu gezeichnet werden
    anfang:=0;           ;//Zurücksetzung in Anfangszustand
  end;

until 2<1;                ;//Schleife wird "unendlich" lang durchlaufen

```



Abb. 2.63

Ein alternatives Vorgehen für die Darstellung mehrerer Graphen in einem Diagramm ist die Möglichkeit einer „mehrfachen“ Abfrage der **Ablauf-Taste** , was besonders für Projekte relevant ist, bei denen die „Endsituation“ (z.B. Vektoren-Anzeigen) interessant ist. Hierfür muss um die bereits vorhandene *repeat*-Schleife eine *begin...end*-Bedingung und eine weitere *repeat*-Schleife gelegt werden, innerhalb derer unterschieden wird, ob die **Ablauf-Taste**  zum Starten oder Zurücksetzen in die Anfangsposition gedrückt wurde. Da dies etwas aufwendiger ist, versuchen Sie die Struktur im folgenden Kernprogramm nachzuvollziehen:

```

C:\PAKMA2000\temp\temp.pkp (editierbar)
Kernprogramm:
// Simulation einer Federschwingung
begin
  repeat
    t:=0;
    startwerte;
    anfang:=0;

    repeat
      //Beginn der physikalischen Berechnungen
      ;
      F:=-k*y;
      a:=F/m;
      if start__=1 then anfang:=1
      if anfang=0 then
        begin
          y:=schieber (ya);
        end
      if anfang=1 then
        begin
          v:=v+a*dt;
          y:=y+v*dt;
          t:=t+dt;
        end;
      ausgabe (t, y, v, a);
    until t>mt;



    repeat
      ausgabe (t, y, v, a);
    until start__=1;

    if start__=1 then anfang:=0;
    neu graf;
  until 2<1;
end;

```



Abb. 2.64

Wie Sie sehen ist diese Struktur etwas komplexer. Zunächst wird nach bereits bekanntem Muster die **Ablauf-Taste**  abgefragt. Nach einmaliger Betätigung ebendieser und Ende der Simulationszeit **mt** erfolgt weiterhin die Ausgabe aller relevanten Größen (vgl. 2. **repeat**-Schleife). Wird danach erneut die **Ablauf-Taste**  gedrückt wird das System in den Anfangszustand (**anfang=0**) zurückgesetzt.

Zum Schluss noch einmal der Hinweis für die Erstellung eines Kernprogramms:

Physikalische Abläufe in die Basisform des Kernprogramms bringen

Auswählen eines Rahmenprogramms

Rahmenprogramm in die Basisform des Kernprogramms einbetten

3 Der graphische Editor

Um Modellbildungen mit einem grafischen Editor durchzuführen, gibt es seit Frühjahr 1999 den visuellen Editor **VisEdit**, mit dem ein Wirkungsgefüge ein physikalisches Modells graphisch erstellt werden kann, aus dem VisEdit dann das zugehörige Kernprogramm für PAKMA automatisch erzeugt. Man kann dabei auf sehr unterschiedliche Weise mit VisEdit und PAKMA arbeiten, im Folgenden wird jedoch nur die Möglichkeit ausgeführt von VisEdit aus ein. Für weitergehende Informationen zum Erstellen von Wirkungsgefüge gibt es das **VisEdit**-Handbuch.

3.1 Vorgehensweise

Im Folgenden wird beschrieben, wie in bereits existierende PAKMA-Projekte mit Hilfe von **VisEdit** Kernprogramme so eingefügt werden können, dass die in PAKMA vorhandenen Startwerte automatisch in das Kernprogramm aufgenommen werden.

Dabei wird vorausgesetzt, dass die beiden benötigten Dateien **FD_SCHW1.prj** und **FD_SCHW1.ved** bereits erstellt sind. Die beiden Dateien finden sich im Ordner **Beispiel**, der sich im gleichen Verzeichnis befindet, wie dieses Handbuch.

In Stichpunkten soll nun erläutert werden, welche Reihenfolge zu beachten ist:

- PAKMA-Projekt **FD_SCHW1.prj** laden

- Wirkungsgefüge **FD_SCHW1.ved** mit Hilfe von VisEdit laden

- Kernprogramm erzeugen

- Programme sichern

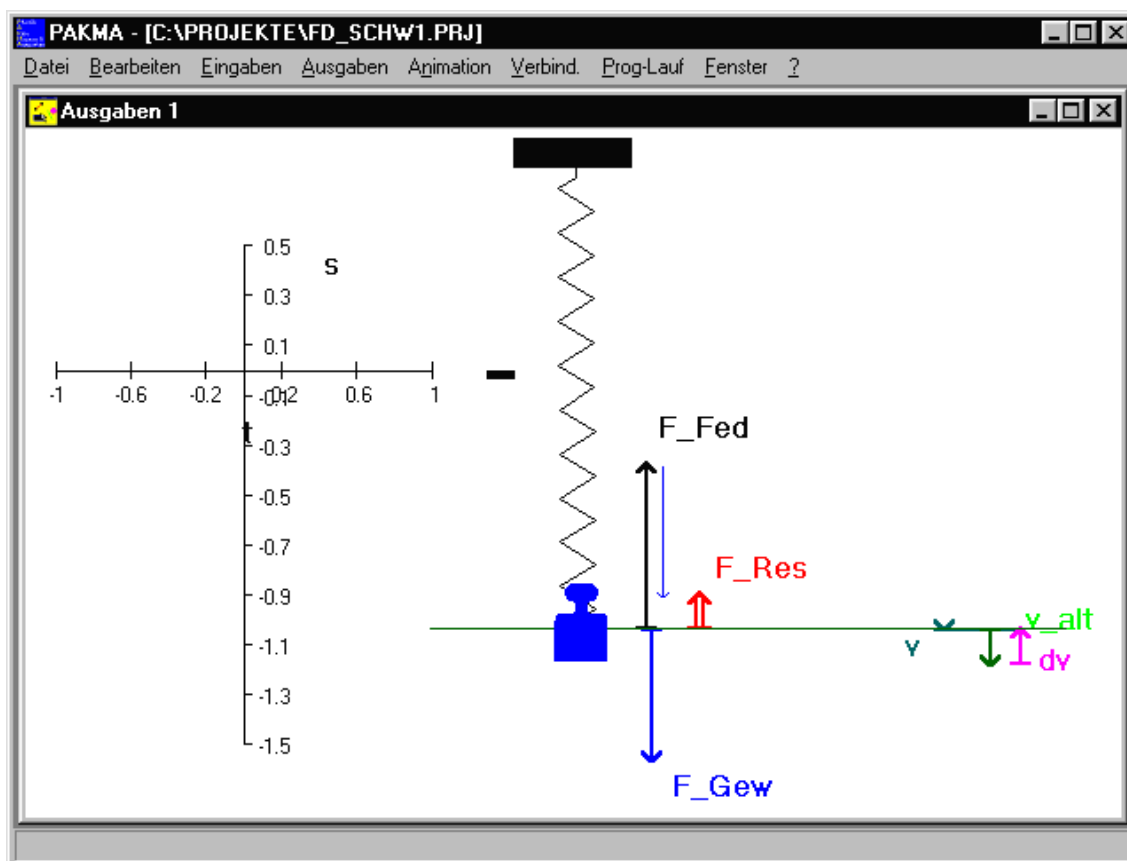


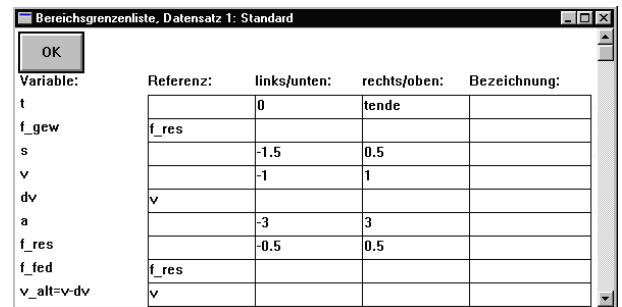
Abb. 3.1

3.2 PAKMA- Projekt laden

Nachdem die Datei **FD_SCHW.prj** geöffnet wurde, sieht die Oberfläche von PAKMA so aus, wie dies **Abb. 3.1** zeigt.

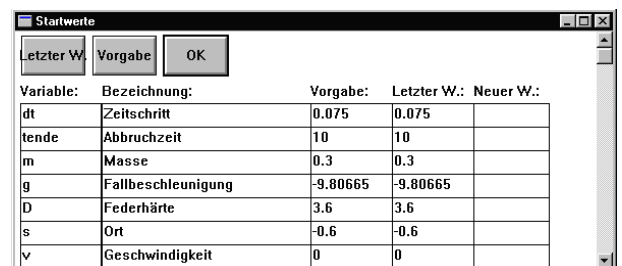
In diesem Projekt sind bereits alle Daten, ausgenommen die des Kernprogramms vorhanden. Die zugehörigen Fenster für die Startwerte und Bereichsgrenzen werden in **Abb. 3.2** und **Abb. 3.3** gezeigt.

Das einzige, was in diesem Projekt noch fehlt, ist das passende Kernprogramm. Wie man unschwer erkennen kann, ist im **Editor** von PAKMA noch kein Eintrag vorhanden.



Variable:	Referenz:	links/unten:	rechts/oben:	Bezeichnung:
t		0	tende	
f_gew	f_res			
s		-1.5	0.5	
v		-1	1	
dv	v			
a		-3	3	
f_res		-0.5	0.5	
f_fed	f_res			
v_alt=v-dv	v			

Abb. 3.2



Variable:	Bezeichnung:	Vorgabe:	Letzter W.:	Neuer W.:
dt	Zeitschritt	0.075	0.075	
tende	Abbruchzeit	10	10	
m	Masse	0.3	0.3	
g	Fallbeschleunigung	-9.80665	-9.80665	
D	Federhärte	3.6	3.6	
s	Ort	-0.6	-0.6	
v	Geschwindigkeit	0	0	

Abb. 3.3

3.3 Wirkungsgefüge laden

Nachdem man **VisEdit** gestartet hat, kann man mit Hilfe des Menüpunktes **Datei** bzw. mit **F3** ein bereits existierendes

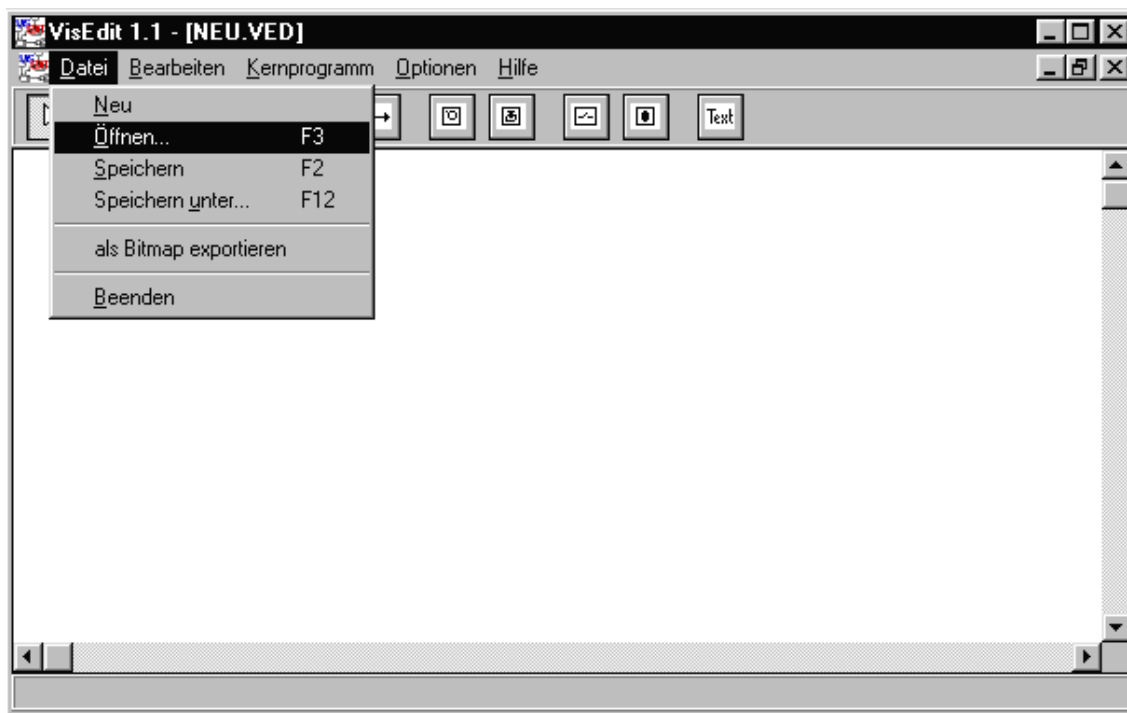


Abb. 3.4

Wirkungsgefüge öffnen. Mit dem **Öffnen** Dialog, kann die jeweils benötigte Datei ausgewählt werden (s. **Abb. 3.5**).

Nachdem die Datei **FD_SCHW1.ved** geöffnet wurde, sieht die Oberfläche von **VisEdit** so aus, wie dies in **Abb. 3.6** zu sehen ist.

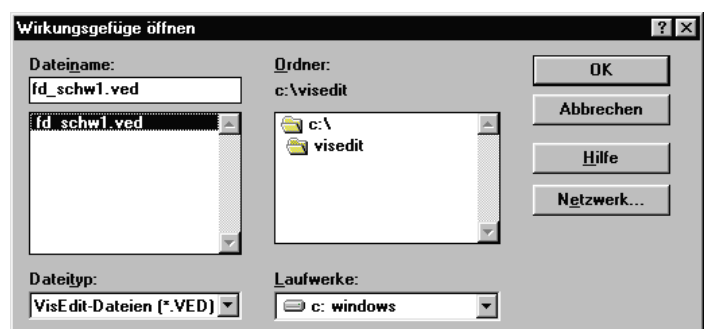


Abb. 3.5

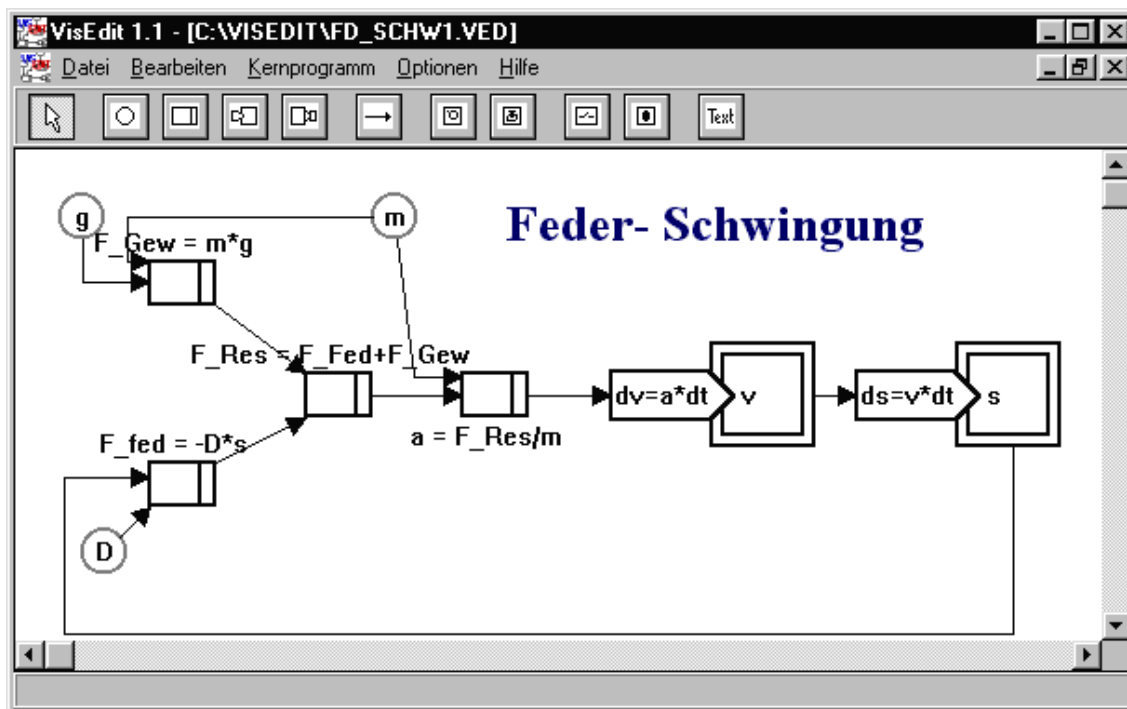


Abb. 3.6

3.4 Kernprogramm erzeugen

Mit Hilfe des Unterpunktes **erzeugen** des Menüpunktes **Kernprogramm** kann nun das benö-

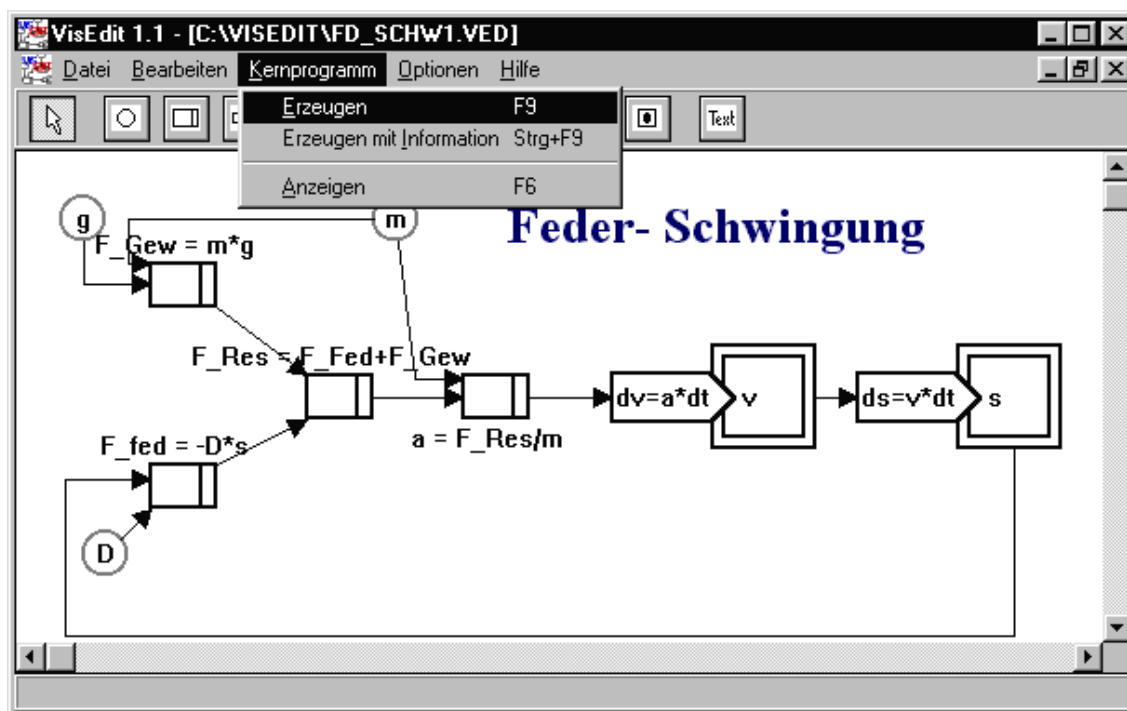


Abb. 3.7

tigte Kernprogramm erstellt werden (s. Abb. 3.7).

Nachdem man diesen Punkt aufgerufen hat, wird das Kernprogramm automatisch in den Editor von PAKMA eingefügt. Dies wird in Abb. 3.8 deutlich, die den Bildschirm mit beiden Programmen zeigt.

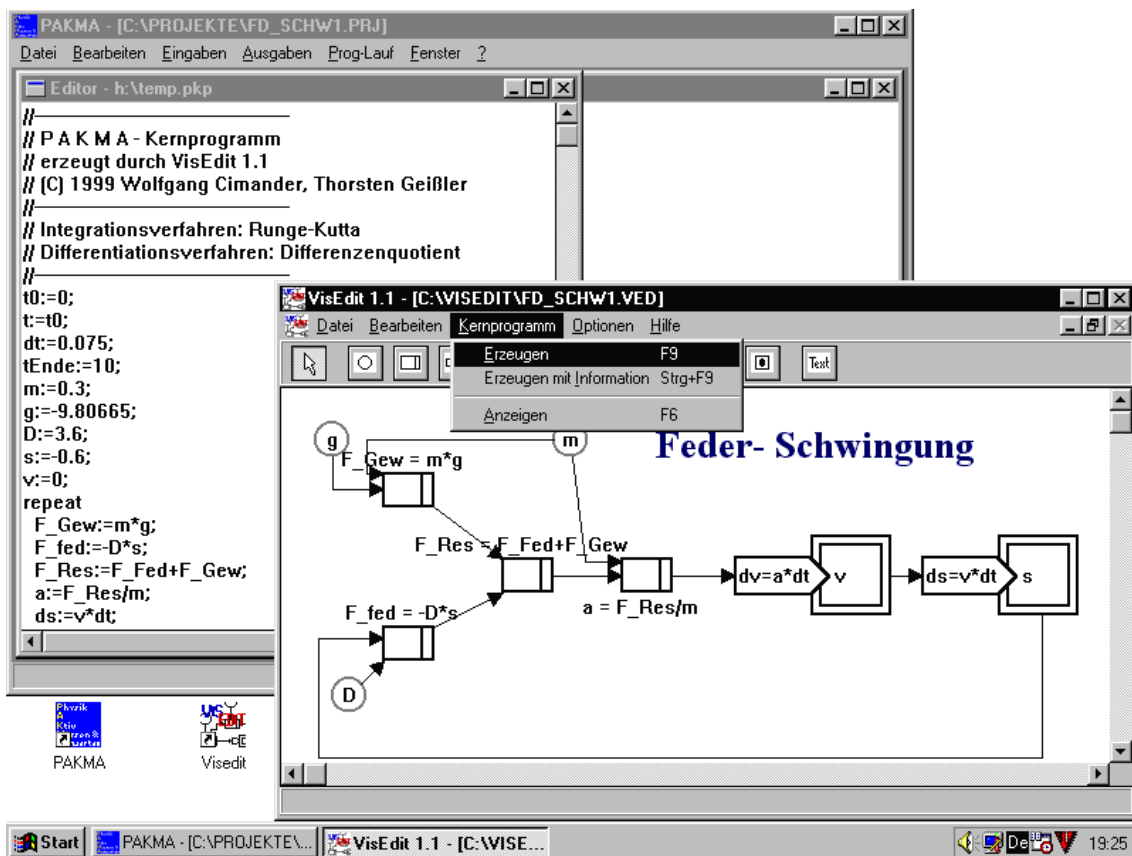


Abb. 3.8

Außerdem kann man in Abb. 3.8 erkennen, dass sämtliche Startwerte am Anfang des Kernprogramms zu finden sind.

Im Programm VisEdit kann man die hierzu nötigen Einstellungen im Unterpunkt **Kernprogrammerzeugung...** des Menü-punktes **Optionen** vornehmen. Dazu steht ein Dialog zur Verfügung, wie er in Abb. 3.9 zu sehen ist.

Man beachte, dass die Funktion **PAKMA-Startwerte übernehmen** zuvor aktiviert wurde.

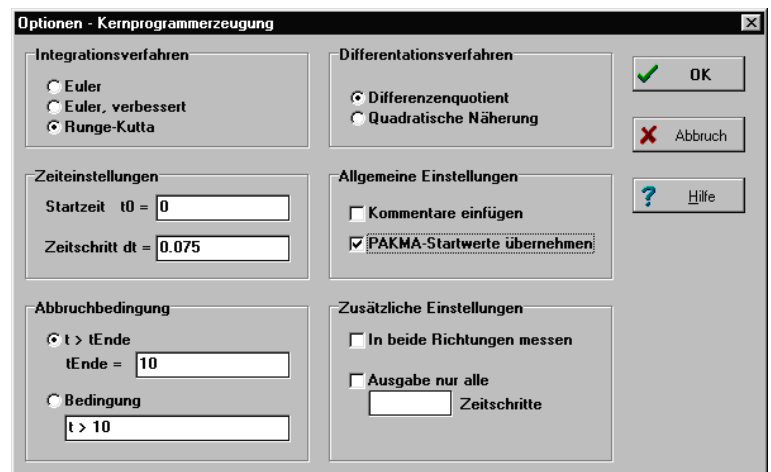


Abb. 3.9

3.5 Programm sichern

Damit das Kernprogramm nicht verloren geht, sollte das jeweilige Projekt abgespeichert werden. Anschließend kann mit PAKMA ein Programmlauf durchgeführt werden.

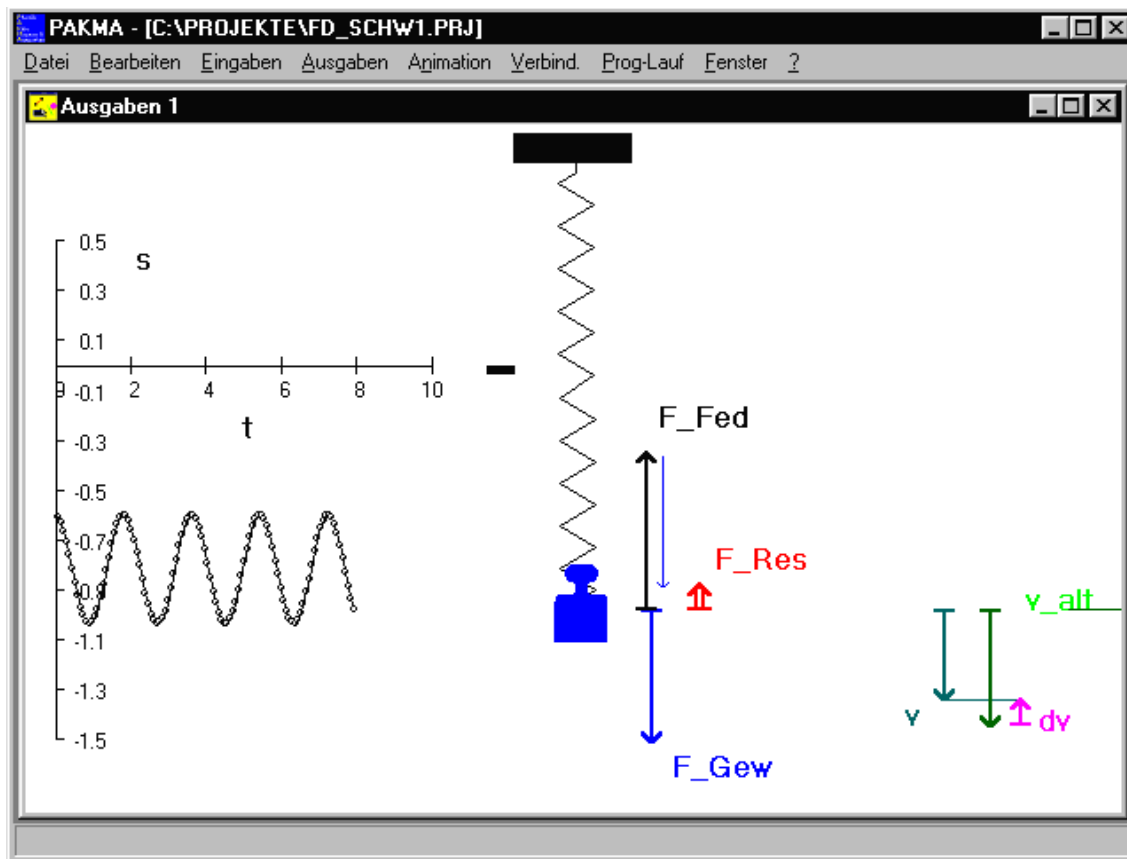


Abb. 3.10

Abb. 3.10 zeigt den PAKMA- Bildschirm, während das Programm **FD_SCHW1.prj** mit Hilfe des neuerstellten Kernprogramms abläuft.

4 Messen mit PAKMA

In diesem Kapitel soll anhand von drei Beispielen erläutert werden, wie man mit PAKMA bei realen Experimenten die Versuchsabläufe erfasst und die erforderlichen Messungen durchführen kann.

Bei der Erstellung eines Projektes, mit dem Messwerte aufgenommen werden sollen, geht man in denselben Schritten vor, wie sie in **Kapitel 3** für ein Modell beschrieben wurden. Der wesentliche Unterschied besteht jedoch darin, dass das Kernprogramm eines Projektes Anweisungen zur Vorbereitung und Durchführung von Messungen enthält. Diese sind erforderlich, da beim Messen mit PAKMA die Aufnahme der Messwerte vom Kernprogramm eines Projektes vorbereitet und ausgeführt wird. Hierzu existieren spezielle PAKMA-Anweisungen, die die Messdatenerfassung steuern und durchführen.

Um mit PAKMA messen zu können, gibt es drei Möglichkeiten:

1. Messungen mit der PAKMA-Interface-Box. Dies ist die universellste Möglichkeit, Messungen durchzuführen. Das Spektrum des Einsatzbereichs reicht von Wegmessungen mit Fahrrad oder Sonarmeter über vielkanalige Spannungsmessungen mit internen Verstärkern bis zur Ansteuerung von Schrittmotoren (s. auch PAKMA-Infoheft)-
2. Low-Cost-Messungen mit der Maus als Wegsensor für 1 und 2-dimensionale Bewegungen, s. Kap. 4.5 und Reusch ...)
3. Messung über externe Messgeräte mit serieller Schnittstelle über COM-Part. Hier können langsame Messungen (typ. 1-5 pro s) mit guter Auflösung z. B. auch mit sehr preiswerten (Lit: Physik in der Schule) Multimetern oder Präzisionswaagen durchgeführt werden.

Zuerst sollen hier Messmöglichkeiten mit der PAKMA-Interface-Box (s. **Abb. 4.1**) vorgestellt werden, in der die Eingangssignale umgesetzt und an eine im Rechner eingesetzte ISA-Einsteckkarte weitergeleitet werden.

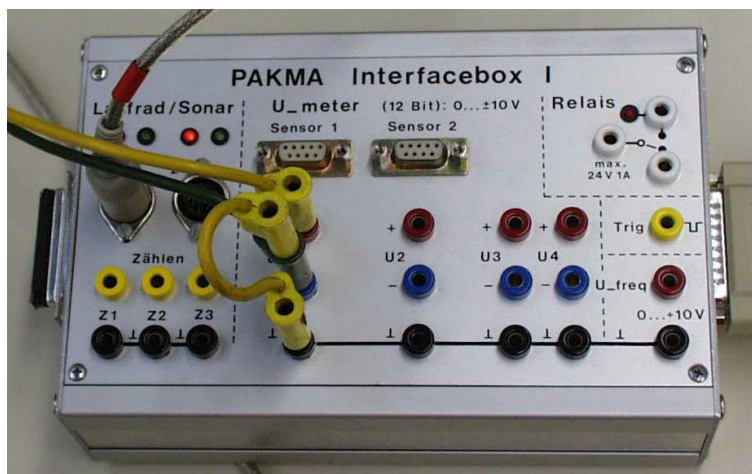


Abb. 4.1

Bevor das Messen mit PAKMA anhand von drei Beispielen vorgestellt wird, sollen die Messbefehle kurz charakterisiert werden.

4.1 Charakterisierung der Messbefehle von PAKMA

Prinzipiell kann man die Messbefehle von PAKMA in die beiden Gruppen Messtypen und Messschritte einteilen.

4.1.1 Messtypen

Mit PAKMA-Befehlen dieser Art wird der Rechner auf das Messen einer bestimmten physikalischen Größe eingestellt. In jedem Fall muss, bevor eine Messung erfolgen kann, ein solcher Messtyp im Kernprogramm festgelegt werden.

Ein einmal aufgerufener Messtyp bleibt so lange aktuell, (d. h. die folgenden Messschritte beziehen sich auf diesen Messtyp) bis er durch einen neu aufgerufenen Messtyp abgelöst wird.

Die Ausführungen in diesem Kapitel werden sich auf die beiden Messtypen

Zählen von Impulsen in einem Zeitintervall sowie

Messen von Spannungen

beschränken, welche nachfolgend kurz vorgestellt werden sollen.

4.1.1.1 Zählen von Impulsen in einem Zeitintervall

Der zugehörige PAKMA-Befehl lautet: **zählen**

Mit dem Messtyp **zählen** werden Impulse innerhalb eines Zeitintervalls **dt** gezählt. Diese Impulszahl kann dann, wie gewünscht, weiterverarbeitet werden. Beispielsweise kann man die Gesamtpulszahl seit Beginn einer Messung ermitteln, indem man die Impulszahlen, die in den einzelnen Zeitintervallen gemessen wurden, durch entsprechende Anweisungen im Kernprogramm aufsummiert.

Mit dem Messtyp **zählen** können Sie maximal auf zwei Kanälen gleichzeitig jeweils zwei Zählarten messen, d. h., es können gleichzeitig vier verschiedene Impulszählungen durchgeführt werden.

Ein Anwendungsgebiet des Messtyps **zählen** ist das Ausmessen von Wegen bzw. Entfernungen. Dies kann realisiert werden, indem pro zurückgelegter Einheitslänge auf geeignete Weise ein Impuls erzeugt wird z. B. mit einer Lichtschranke. Mit einer Doppelgabellichtschranke kann auch eine Vor-rückwärts-Bewegung erfasst werden. Die Zählimpulse, die die Rückwärtsbewegung erfassen, werden dann als zweite Zählrate gemessen. Multipliziert man dann die Anzahl der Impulse, welche in einem Zeitintervall **dt** registriert und vorzeichengerecht addiert wurden, mit der Einheitslänge, so erhält man den während dieses Zeitintervalls zurückgelegten Weg. Ein Anwendungsbeispiel hierfür ist die Aufnahme der Ort-Zeit-Kurve des Federschwingers. Dieses Beispiel wird in **Abschnitt 4.2** ausführlich erläutert.

4.1.1.2 Spannungen messen

Der zugehörige PAKMA-Befehl lautet: **u_meter**

Mit Hilfe dieses Messtyps können Spannungen gemessen werden. Die maximalen Werte, die dabei vom Rechner erfasst werden können, sind ± 10 Volt. Demzufolge müssen, wenn nötig, die aufzunehmenden Spannungen über eine Spannungsteilerschaltung so verkleinert werden, dass sie vom Rechner gemessen werden können. Als Meßbereiche stehen $\pm 10V$, $\pm 1V$, $\pm 0,1V$ sowie $\pm 0,01V$ zur Verfügung.

Mit dem Messtyp **u_meter** kann gleichzeitig auf bis zu 16 Kanälen gemessen werden. Ein Beispiel für den Einsatz des Messtyps **u_meter** ist die Untersuchung von Auf- und Entladevorgängen am Kondensator, welches in **Abschnitt 4.3** vorgestellt wird.

4.1.2 Messschritte:

Nachdem ein bestimmter Messtyp festgelegt wurde, wird die Messung durch die Angabe sogenannter Messschritte gesteuert und durchgeführt. Erst durch diese Messschritte wird der Messvorgang ermöglicht.

Messschritte können prinzipiell erst dann ausgeführt werden, wenn zuvor ein Messtyp aufgerufen wurde. Alle folgenden Messschritt-Befehle beziehen sich dann auf diesen Messtyp.

Welche Messschritte zur Verfügung stehen, und in welcher Reihenfolge diese anzuwenden sind, wird anhand der oben genannten Beispiele dargelegt.

4.2 Das Ort-Zeit-Verhalten eines Federschwingers

In diesem Abschnitt soll am Beispiel eines Federschwingers erläutert werden, wie man mit PAKMA Wegmessungen durchführen kann. Dieses Beispiel bietet sich als Einführung zum Messen an, da hier das Kernprogramm die einfachste Struktur aufweist, die allgemein bei Messprogrammen möglich ist.

Wie bereits erwähnt, können Wegmessungen mit dem Messtyp **zählen** realisiert werden, wobei für die Messwert-Erfassung ein Zählrad verwendet wird.

Ziel dieses Versuches ist es, die Auslenkung x , die Geschwindigkeit v und die Beschleunigung a in Abhängigkeit von der Zeit t zu betrachten und mit PAKMA graphisch in einem Diagramm darzustellen. Dazu soll die Auslenkung x bei der Durchführung des Versuches mit PAKMA gemessen werden. Die anderen Größen (v und a) können dann einfach aus x berechnet werden.

Für die Durchführung des Versuches wurde folgender Aufbau verwendet:

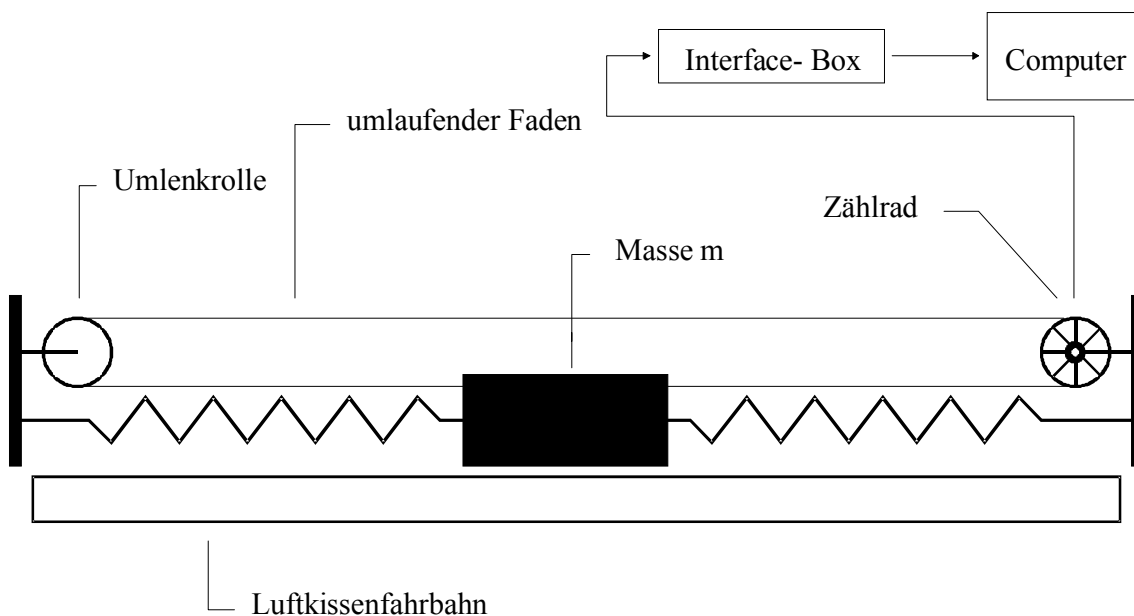


Abb. 4.2

Eine an zwei Federn befestigte Masse, die auf einer Luftkissenbahn gleitet, wird mittels eines umlaufenden Fadens mit einem Zählrad verbunden. Dieses Zählrad wird über eine Interface-Box an den Computer angeschlossen. Möchte man ein Kernprogramm zum Messen mit PAKMA erstellen, so ist festzustellen, dass ein solches Messprogramm dieselbe Struktur wie das Kernprogramm eines Modells besitzt. Auch hier sind am Anfang die benötigten Startwerte und Konstanten zu definieren.

Die wichtigste Festlegung ist die Angabe des Messtyps. Damit wird der Rechner auf die zu messende Größe eingestellt und die nachfolgenden Messschrittbefehle beziehen sich alle auf diesen Messtyp. Wie bereits erläutert kann man mit PAKMA Wegmessungen durch den Messtyp „Zählen von Impulsen in einem Zeitintervall“ realisieren. Die zugehörige PAKMA-Anweisung lautet: *zählen*

zählen;

Nun sind als weitere Messbefehle die Messschritte anzugeben. Die Messschritte von PAKMA sind einfach zu erlernen, da sie im Wesentlichen den Handlungen entsprechen, welche bei manuell durchgeführten Messungen auszuführen sind. Bei einigen Messschrittbefehlen müssen zusätzliche Parameter angegeben werden, dies wird bei den entsprechenden Befehlen näher erläutert.

Bevor der eigentliche Messvorgang, der dem Ablesen eines Meßgerätes entspricht, erfolgen kann, sind einige Vorbereitungen zu treffen.

Zuerst erfolgt der Messschritt „vorbereiten“. Damit wird der Rechner auf das Aufnehmen einer neuen Messreihe vorbereitet. Der zugehörige PAKMA-Befehl lautet **vorb(f)**, wobei als Parameter eine Feldvariable **f** angegeben werden muss, die angibt, in welchem Feld die Messwerte gespeichert werden sollen. Dies ist notwendig, um Messwerte gegebenenfalls später bei einer Reproduktion wieder aus dem Speicher aufrufen zu können. Es stehen dazu 16 Messdatenfelder zur Verfügung, die mit 1, 2, ... durchnummeriert sind und dadurch voneinander unterschieden werden können.

Die PAKMA-Anweisung für das Kernprogramm lautet also:

vorb(1);

Nun legt man die äußere Form des Messablaufs durch den Befehl „Schrittfolge“ fest. Mit diesem Befehl gibt man an, auf welche Art und Weise zwei Messungen nacheinander durchgeführt werden sollen.

Für unser Beispiel bietet sich die zeitgesteuert intermittierende Messung an. Bei dieser Art der Schrittfolge wird bei Aufruf der Messung (s. u.: Befehl *mes*) vor dem „Ablesen“ des Messwertes so lange gewartet, bis ein vorgegebenes Zeitintervall Δt ab dem letzten Meßaufruf abgelaufen ist. Genau dann wird die erneute Messung ausgeführt.

In unserem Beispiel lautet also die Anweisung für die Schrittfolge:

s_folg('i', Δt);

Dabei steht der erste Parameter 'i' für die durchzuführende zeitgesteuert intermittierende Messung, der zweite Parameter Δt gibt das Zeitintervall zwischen zwei Messungen in Sekunden an. Dieses Zeitintervall Δt definiert man als Konstante zu Beginn des Kernprogramms. Für unser Beispiel wählen wir: **$\Delta t := 0.1$;**

Da bei dieser Art der Messung nach dem Ausführen des Messbefehls zum nachfolgenden Befehl im Kernprogramm übergegangen wird, besteht die Möglichkeit, jeweils zwischen zwei Messungen die Aufbereitung der Messwerte (eventuell notwendige Berechnungen sowie die Darstellung der Ausgabegrößen in einem Ausgaben-Fenster) vorzunehmen. Der Verlauf der Ausgabegrößen kann somit parallel zum Versuchsablauf auf dem Bildschirm beobachtet werden.

Da das Federpendel sowohl nach links wie auch nach rechts schwingt, werden die in Δt gezählten Impulsraten entsprechend ihrem positivem bzw. negativem Vorzeichen bereits intern addiert. Dazu benötigt man im Kernprogramm die Anweisung

```
vor_rück;
```

Die Vorbereitungen und notwendigen Festlegungen zur Durchführung der Messung sind mit den soeben erläuterten Anweisungen getroffen. Nun kann man dazu übergehen, die Messwerte für die Auslenkung x zu erfassen. Dies erfolgt innerhalb einer **REPEAT-UNTIL**-Schleife über die Messzeit. Zwischen der Aufnahme von zwei Messwerten wird die Messung ausgewertet. Dies beinhaltet, dass die nicht durch Messung erfassten Ausgabegrößen berechnet und die Zahlenwerte der Ausgabegrößen durch die Ausgabeanweisung zur Darstellung im Ausgaben-Fenster bereitgestellt werden. Die Im Folgenden erläuterten Anweisungen sind innerhalb der Messschleife auszuführen.

Der eigentliche Befehl „Messung ausführen“, welcher dem Ablesen eines Meßgerätes entspricht, ist für unser Beispiel sehr einfach. Da nur eine Größe, nämlich die Auslenkung x , gemessen werden soll, spricht man von einer einkanaligen Messung. Die Anweisung zur Messung lautet für diesen Fall:

```
mes;
```

Den aufgenommenen Messwert ruft man bei einer einkanaligen Messung mit der Anweisung `mes_w` auf. Dieser Messwert muss dann einer Variable im Kernprogramm zugewiesen werden. Bei dem Messtyp zählen wird beim Aufruf des Messwertes eine Zahl geliefert, welche der Anzahl der Impulse entspricht, die in dem zuletzt abgelaufenen Zeitintervall dt registriert wurden. Um daraus die Änderung der Auslenkung dx seit der letzten Messung berechnen zu können, benötigt man die Länge des Weges, welcher zwischen der Registrierung von zwei Impulsen zurückgelegt wurde. Hierbei handelt es sich um einen konstanten Faktor, der bei Verwendung eines Zählrades mit `l_einh` bezeichnet wird und für das Präzisionslaufrad den Wert 0.000804 hat. Dieser Faktor wird zu Beginn des Kernprogramms als Konstante definiert:

```
l_einh:=0.000804;
```

Die Änderung der Auslenkung während des Zeitintervalls dt , welche mit dx bezeichnet ist, kann nun über folgende Anweisung berechnet werden:

```
dx:= mes_w * l_einh;
```

Anschließend erhält man die gesamte Auslenkung x durch die Beziehung:

```
x:=x+dx;
```

Die Geschwindigkeit v und die Beschleunigung a können dann durch einfache Rechnung aus x und dx bestimmt werden: $v:=dx/dt$;

```
dv:=v-v_alt;
```

```
a:=dv/dt;
```

Da diese Rechnung leicht nachvollziehbar ist, soll sie nicht näher erläutert werden.

Natürlich müssen für x und v_alt zu Beginn des Kernprogramms Anfangswerte definiert werden: $x:=0$;

```
v_alt:=0;
```

Nun können durch den Ausgabebefehl die Ausgabegrößen festgelegt und zur Darstellung in einem Ausgaben-Fenster bereitgestellt werden:

```
ausgabe (t,x,v,a);
```

Zum Abschluss müssen die Werte für t und v_alt aktualisiert werden. Dies geschieht durch folgende Anweisungen: $t:=t+dt$;

```
v_alt:=v;
```

Damit sind alle erforderlichen Anweisungen für das Kernprogramm dieses Projektes getroffen und die Messschleife kann erneut durchlaufen werden.

Das gesamte Kernprogramm, welches in das Editor-Fenster eingegeben wird, lautet:

```
t:=0; dt:=0.1; mt:=10; // Anfangswert und Konstanten
l_einh:=0.0009; // Weglänge zwischen zwei Impulsen
v_alt:=0; // Anfangswerte
x:=0;
;
zählen; // Festlegen des Messtyp
vorb(1); // Messvorgaben
s_folg('i',dt);
vor_rück;
;
REPEAT // Messschleifenbeginn
mes; // Messung
dx:=mes_w * l_einh; // Berechnungen der Ausgabegrößen
x_neu:=x+dx;
v_neu:=dx/dt;
v:=(v_neu-v_alt)/2
dv:=v_neu-v_alt;
a:=dv/dt;
ausgabe (t,x,v,a); // Ausgabeanweisung
t:=t+dt;
x:=x_neu;v_alt:=v_neu;
UNTIL t>mt; // Abbruchbedingung der Messschleife
```

*An dieser Stelle sei erwähnt, dass die PAKMA-Befehle **vorb(f)**, **s_folg(...)**, **mes** und **mes_w** grundlegende Messschritte sind, die unabhängig vom Messtyp in dieser oder abgewandelter Form in der Regel in jedem Messprogramm vorkommen müssen. Erscheint z. B. der Befehl **s_folg(...)** nicht, so wird bei Aufruf des Messbefehls **mes** lediglich eine Einzelmessung durchgeführt. Dies ist jedoch nur in speziellen Situationen anzuwenden (s. **Abschnitt 4.4**).*



Bei der weiteren Ausarbeitung des Projektes fährt man ebenso fort, wie es im **Kapitel 3** beschrieben wurde. Man geht also im nächsten Schritt zur Wahl der Bereichsgrenzen über. Hierzu sei noch angemerkt, dass man die Bereichsgrenzen gegebenenfalls durch Probieren bestimmen kann.

Die Gestaltung des Ausgaben-Fensters sowie das Erstellen einer dynamischen Repräsentation des Vorgangs erfolgen ebenso, wie für ein Modell.

Bevor man das neu erstellte Projekt ablaufen lassen kann, ist es sinnvoll, dieses zunächst über den Menüpunkt **Datei → Speichern unter** unter einem geeigneten Namen abzuspeichern.

Möchten man das Projekt nun starten und dabei eine neue Messreihe aufnehmen, so ruft amn aus dem Menüpunkt **Prog-Lauf** die Startform **Start als Original** auf. Nachdem das Kernprogramm compiliert wurde erscheint eine Meldung auf dem Bildschirm, die besagt, dass das Kernprogramm nun startbereit ist. Durch Anklicken der Schaltfläche **OK** erfolgt der Start des Kernprogramms. Das reale Experiment wird kurz danach aus der Ruhelage gestartet. Während

der Abarbeitung des Kernprogramms werden dann die gewünschten Messwerte mit dem Rechner erfasst. Dabei wartet der Rechner, bevor eine neue Messung erfolgt, bis das gewählte Zeitintervall Δt abgelaufen ist. Die Aufnahme der Messwerte endet, wenn die vorgesehene Messzeit überschritten wird.

Damit die erfassten Messwerte nicht verloren gehen, sollten Sie diese nach Ablauf des Projektes zunächst über den Menüpunkt **Datei** → **Modul speichern** → **Messdaten** unter einem geeigneten Namen abspeichern. Andernfalls werden sie bei einem weiteren Start durch die neuen Messwerte überschrieben.

Möchten Sie mehrere verschiedene Messreihen aufnehmen, so bietet es sich an, vor einem erneuten Start das Messdatenfeld **f** im Messschrittbefehl **vorb(f)** zu wechseln. Dadurch bleiben die letzten Messwerte erhalten und die neuen Messwerte werden in einem anderen Messdatenfeld abgelegt. Die verschiedenen Messreihen können dann zusammen unter einem Namen abgespeichert werden.

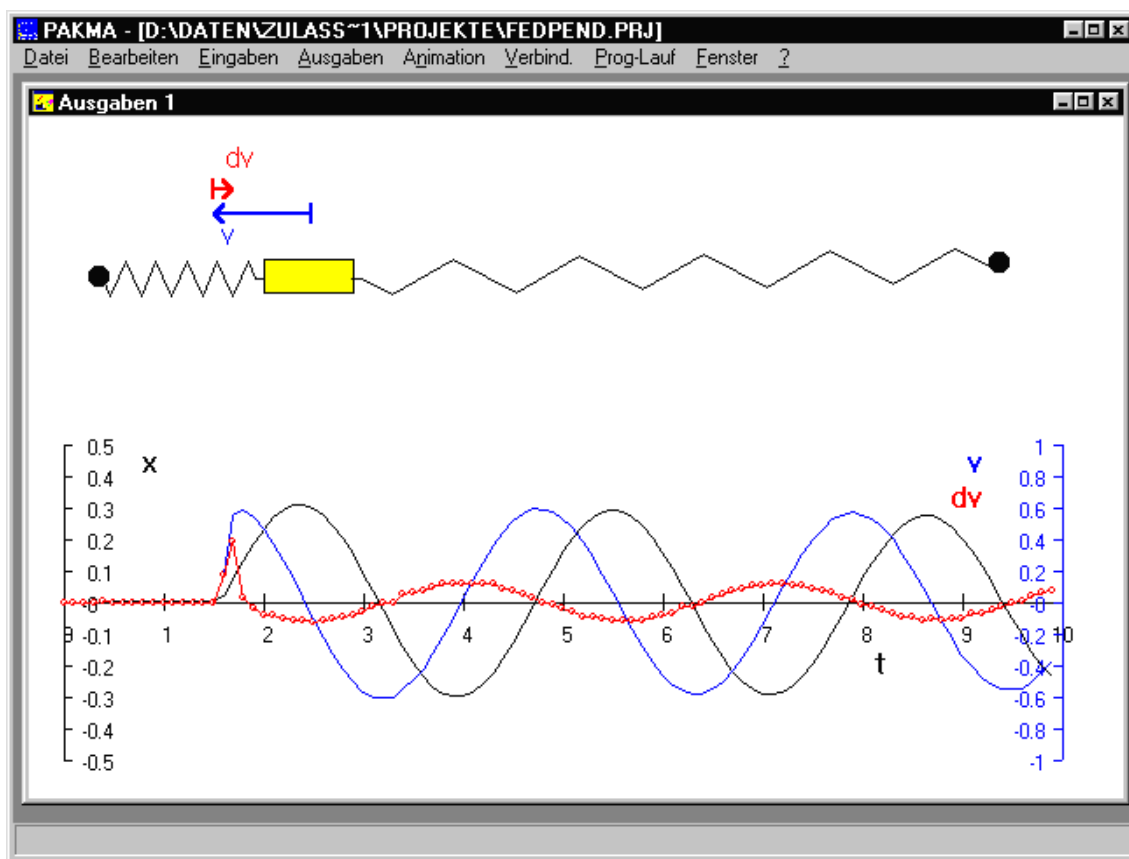


Abb. 4.3

Am einfachsten und übersichtlichsten kann man das Wechseln der Messdatenfelder durch Verwenden des Fensters Startwerte erreichen.

Abb. 4.3 zeigt ein mögliches Versuchsergebnis.

Das Projekt kann natürlich beliebig erweitert werden, indem man zusätzliche Größen, wie z.B. die potentielle bzw. kinetische Energie, berechnet und diese ebenfalls graphisch in einem Diagramm darstellt. Dies kann sehr einfach über den Menüpunkt **Eingaben** → **Variablen definieren** → **Neue Variable definieren...** realisiert werden.

Eine weitere Möglichkeit der Erweiterung ergibt sich, wenn man zusätzlich Modell-berechnungen einfügt, um einen direkten Vergleich zwischen Experiment und Modell zu erhalten. Eine derartige Modellierung eines Realexperimentes wird im folgenden Beispiel vorgestellt.

4.3 Ladungsvorgänge am Kondensator

Ziel dieses Versuches ist es, den zeitlichen Verlauf von Auf- und Entladevorgängen am Kondensator zu untersuchen. Hierzu sollen vom Rechner über den Messtyp **u_meter** mehrere Spannungen gleichzeitig gemessen werden. Außerdem wird eine Modellierung der Ladungsvorgänge durchgeführt, wodurch ein direkter quantitativer Vergleich zwischen Experiment und Modell ermöglicht wird.

Die Ausführungen in diesem Abschnitt beschäftigen sich schwerpunktmäßig mit den Veränderungen innerhalb des Kernprogramms, welche sich im Vergleich zu dem Beispiel aus Abschnitt 4.2 durch die mehrkanalige Messung mit dem Messtyp **u_meter** sowie die vorgenommene Modellierung ergeben.

Genaue Ausführungen zum Aufbau und zur Durchführung des Versuches, sowie zur Auswertung und Aufbereitung der Messwerte findet man in der Zeitschrift Praxis der Naturwissenschaften – Physik2.

Bei der Versuchsdurchführung wurde folgende Schaltung verwendet, welche der obengenannten Zeitschrift entnommen wurde:

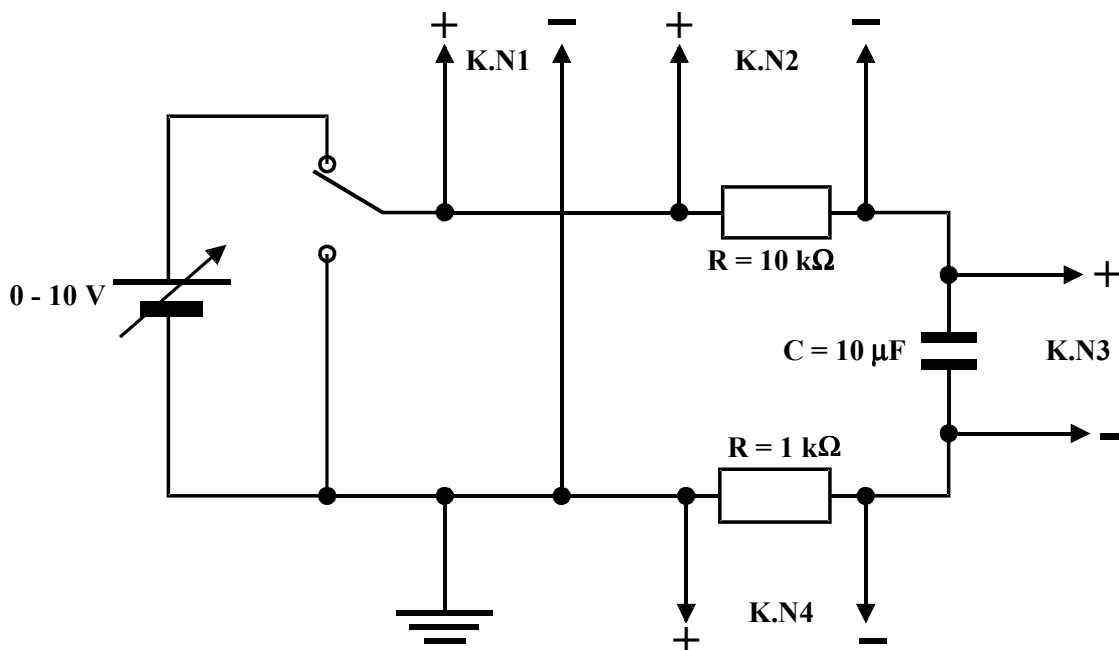


Abb. 4.4

Mit dieser Schaltung werden die Ladevorgänge am Kondensator über das Rechner-Interface erfasst. K.N1 usw. bezeichnen die Kanal-Nummer des Spannungseingangs an der Interface-Box.

Wie bereits im **Abschnitt 4.1** erläutert, kann man mit dem Messtyp **u_meter** gleichzeitig auf bis zu 16 Kanälen messen. In unserem Beispiel erfolgt eine mehrkanalige Messung auf 4 Kanälen, wobei folgende Größen vom Rechner erfasst werden sollen (s. **Abb. 4.4**):

Kanal 1: Betriebsspannung U_B

Kanal 2: Spannungsabfall am Vorwiderstand U_R

Kanal 3: Spannungsabfall am Kondensator U_C

Kanal 4: Spannungsabfall am Messwiderstand U_{MW} , der zur Berechnung des Stromes I verwendet wird

Diese Größen können später in ihrem zeitlichen Verlauf auf verschiedene Weisen im Ausgaben-Fenster veranschaulicht bzw. zur Berechnung weiterer Größen (z. B. des Stromes I) verwendet werden.

Das vollständige Kernprogramm zu diesem Projekt findet man auf Seite 107 der zuvor erwähnten Zeitschrift.

Zu Beginn werden wiederum die benötigten Startwerte und Konstanten definiert. Im Anschluss daran legt man den zu verwendenden Messtyp fest:

```
u_meter;
```

Nun folgt, analog zum ersten Beispiel, der Messschritt `vorb(f)`. Bei n -kanaliger Messung werden die auf den verschiedenen Kanälen gemessenen Größen jeweils in einem Messdatenfeld abgelegt. Der Parameter f gibt für diesen Fall die Anfangsnummer der verwendeten Messdatenfelder an. Es stehen jetzt nur weniger Messdatenfelder zur Verfügung; bei einer vierkanaligen Messung nur $16/4 = 4$.

Da eine zeitgesteuert intermittierende Messung erfolgen soll, lautet der Messschritt „Schrittfolge festlegen“ ebenso wie im ersten Beispiel:

```
s_folg ('i',dt);
```

Das Zeitintervall dt zwischen zwei Messungen wird auch hier als Konstante zu Beginn des Kernprogramms definiert.

Weiterhin muss man bei Verwendung des Messtyps `u_meter` festlegen, in welchen Messbereichen die Spannungsmessungen auf den verschiedenen Kanälen erfolgen sollen. Die Anweisung, welche dies angibt, lautet `m_bereich(b,k)`, wobei b die vorgesehene Bereichseinstellung in Volt und k die zugehörige Kanalnummer 1,...,16 bestimmt. Für den Parameter b (Bereichseinstellung) stehen Ihnen folgende Möglichkeiten zur Verfügung:

Parameter b	verwendeter Meßbereich
10	-10 bis +10 V
1	-1 bis +1 V
0.1	-0,1 bis +0,1 V
0.01	-0,01 bis +0,01 V

Gibt man für die Kanalnummer k die Zahl 0 ein, so gilt die festgelegte Messbereichs-einstellung für alle Kanäle.

In unserem Beispiel soll für die Kanäle 1 bis 3 der Messbereich 10 verwendet werden, für den Kanal 4 benötigt man den Meßbereich 0.1. Somit lauten die Befehle für die Messbereichseinstellung:

```
m_bereich(10,0);
```

```
m_bereich(0.1,4);
```

Hierbei ist es wichtig, dass die beiden Befehle in der angegebenen Reihenfolge im Kernprogramm erscheinen. Bei umgekehrter Reihenfolge würde die Anweisung `m_bereich(10,0)` auch für Kanal 4 gelten.

Danach beginnt die Messschleife (REPEAT-UNTIL-Schleife über die Meßzeit), innerhalb welcher die mehrkanalige Messung, die Berechnung des Stromes I sowie die Modellierung erfolgen.

Als erstes folgt der Meßschritt „Messung durchführen“. Bei Parallelmessung auf mehreren Kanälen lautet der zugehörige PAKMA-Befehl `mes_p(k)`, wobei k die Anzahl der Kanäle angibt, auf welchen gemessen werden soll. In unserem Beispiel heißt die Meßanweisung:

```
mes_p(4);
```

Nun können die Messwerte der einzelnen Kanäle durch den Befehl **mes_wp(k)** aufgerufen und zugewiesen werden. Der Parameter **k** gibt hierbei die entsprechende Kanalnummer an. Damit ergeben sich folgende Anweisungen für das Kernprogramm:

```
ub:=mes_wp(1);
```

```
ur:=mes_wp(2);
```

```
uc:=mes_wp(3);
```

```
u_mw:=mes_wp(4);
```

Im Anschluss daran erfolgt die Berechnung des Stromes **I** aus dem Spannungsabfall am Meßwiderstand nach dem Ohmschen Gesetz:

```
i:=u_mw/rm;
```

Die Größe des Meßwiderstandes wird zu Beginn des Kernprogramms als Konstante definiert:

```
rm:=...;
```

Bei der nun folgenden Modellierung des physikalischen Sachverhaltes, werden der Strom **I** sowie der Spannungsabfall **UC** am Kondensator betrachtet. Dabei sollen die Modellberechnungen beginnen, sobald bei Ablauf des Projektes die Taste „m“ der Tastatur gedrückt wird, wobei als Startwerte die momentanen Messwerte **UC** und **I** übergeben werden.

Dies kann durch folgende if-then-Anweisung realisiert werden:

```
if taste 'm' then
```

```
begin
```

```
mod_an:=1;
```

```
uc_mod:=uc; ic_mod:=i;
```

```
end;
```

Hierbei wird im Unterschied zu den experimentellen Größen an die entsprechenden modellmäßig berechneten Größen die Endung „_mod“ angehängt. Durch die Anweisung **mod_an:=1** wird die Modellierung „eingeschaltet“, die Modellberechnungen beginnen dann zu diesem Zeitpunkt. Zuvor gilt **mod_an:=0**, was zu Beginn des Kernprogramms festgelegt wird.

Ist die Modellierung eingeschaltet, so erfolgen bei jedem erneuten Schleifendurchlauf die Modellberechnungen von **UC** und **IC** innerhalb folgender if-then-Anweisung:

```
if mod_an:=1 then
```

```
begin
```

```
ic_mod:=(ub-uc_mod)/r;
```

```
uc_mod:=uc_mod+ic_mod*dt/c;
```

```
end;
```

Die Größen für **C** und **R** sind ebenfalls als Konstanten festzulegen.

Weiterhin sei angemerkt, dass die Reihenfolge der beiden if-then-Anweisungen entscheidend ist. Im Kernprogramm muss zuerst die Modellberechnung erscheinen, andernfalls wäre die Modellierung zur Messung zeitlich verschoben.

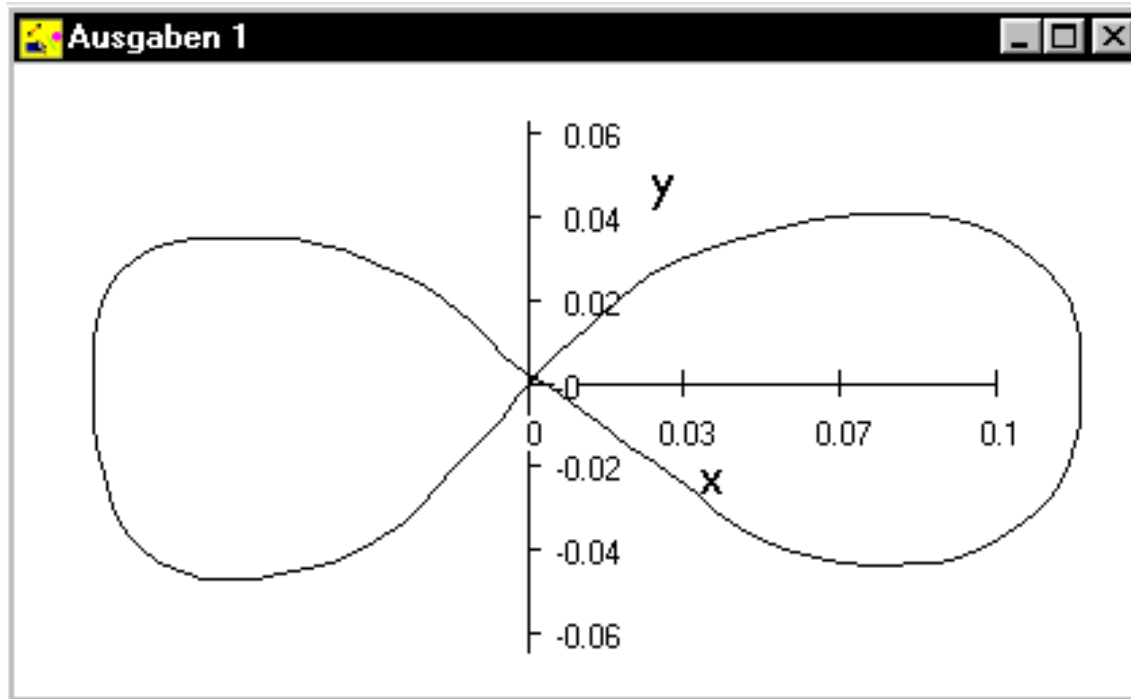


Abb. 4.5

Es folgt nun die Ausgabeanweisung des Kernprogramms, in welcher alle Größen aufgeführt werden, welche in einem oder mehreren Ausgaben-Fenstern veranschaulicht werden sollen:

```
ausgabe (t,ub,uc,ur,i,uc_mod,ic_mod);
```

Zum Abschluss wird innerhalb der Messschleife die Zeit t um das als Konstante festgelegte Zeitintervall dt erhöht:

```
t:=t+dt;
```

Man erhält somit folgendes Kernprogramm:

```
t:=0;mod_an:=0;           // Startwerte
dt:=0.1;mt:=40;           // Messintervall und -zeit
rm:=1000;                 // Messwiderstand in Ohm
c:=2.1E-5;                // C in F
r:=100000;                // R in Ohm
abl:=1;                   // Konstante
;
u_meter;                  // Messvorgaben
vorb(abl);
s_folg('i',dt);
m_bereich(10,0);m_bereich(0.1,4);
;
REPEAT                    // Schleifenbeginn
  mes_p(4);               // Messung
  ub:=mes_wp(1);
  ur:=mes_wp(2);
  uc:=mes_wp(3);
  u_mw:=mes_wp(4);
```

```

i:=u_mw/rm;           // Berechnung des Stromes I
;
if mod_an:=1 then      // Modell-Anfang
begin
    ic_mod:=(ub-uc_mod)/r;
    uc_mod:=uc_mod+ic_mod*dt/c;
end;                  // Modell-Ende
;
if taste 'm' then      // bei Tastendruck von 'm'
begin
    mod_an:=1;         // Modell "einschalten"
    uc_mod:=uc; ic_mod:=i; // Startwerte zuweisen
end;
;
ausgabe(t,ub,uc,ur,i,uc_mod,ic_mod); // Ausgabeanweisung
t:=t+dt;
UNTIL t>mt;           // Abbruchbedingung der Messschleife
    
```

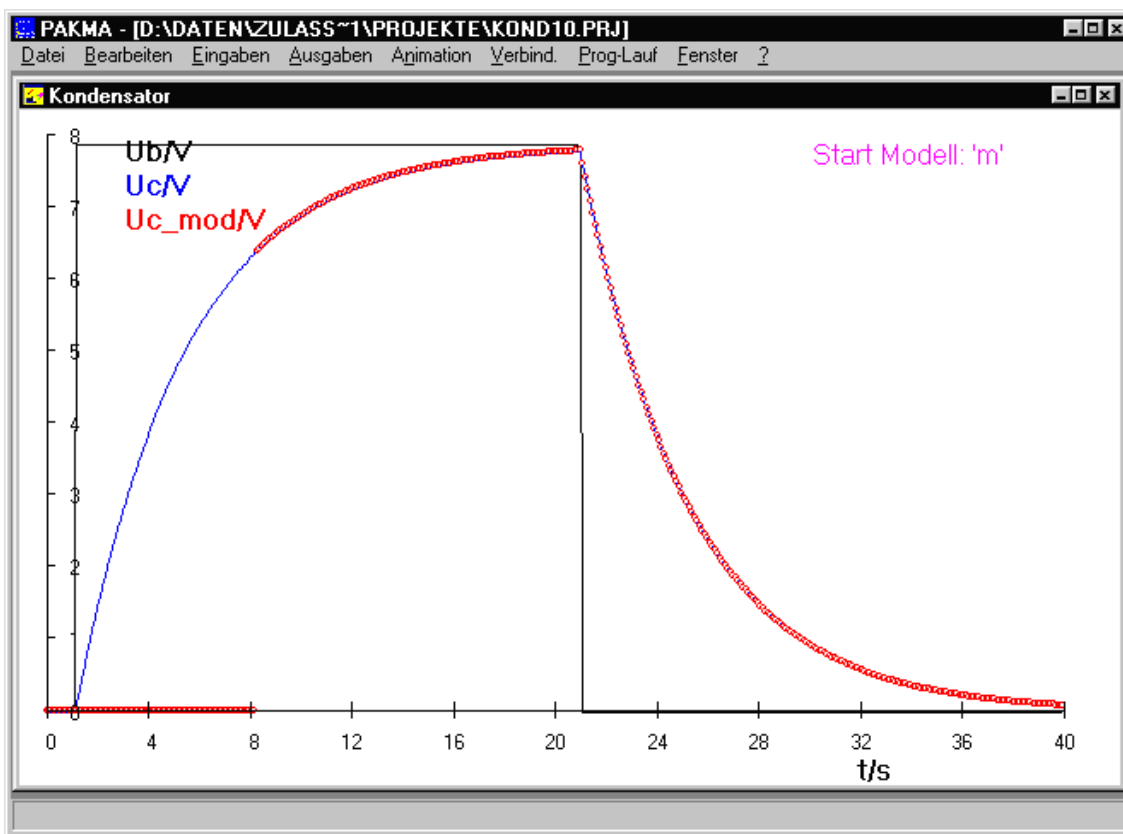


Abb. 4.6

Nachdem das Kernprogramm erstellt wurde, geht man bei der weiteren Entwicklung des Projektes ebenso vor, wie es in **Kapitel 3** für ein Modell beschrieben wurde.

Natürlich kann man das Projekt beliebig erweitern, indem man zusätzliche Größen, wie z. B. die Ladung Q , aus den gemessenen Größen berechnet. **Abb. 4.6** zeigt ein mögliches Versuchsergebnis des Projektes.

4.4 Magnetfeld längs der Achse von Kreisströmen

Anhand dieses Beispiels soll vorgestellt werden, wie Sie mit PAKMA gleichzeitig Messungen mit zwei verschiedenen Messtypen durchführen können. Ziel dieses Versuches ist es, das Magnetfeld einer Kreisspule in Abhängigkeit vom Ort zu messen. Dabei wird das Magnetfeld mittels einer Hallsonde als Spannung über den Messtyp **u_meter**, der Ort durch ein Zählrad über den Messtyp **zählen** erfasst.

In diesem Abschnitt sollen kurz der Aufbau des Versuches, schwerpunktmäßig jedoch die Besonderheiten des zugehörigen Kernprogramms, welche sich durch das Verwenden von zwei verschiedenen Messtypen ergeben, vorgestellt werden. Genaue Ausführungen zur Versuchsanordnung, Durchführung und Auswertung finden Sie in der Zeitschrift „Praxis der Naturwissenschaften – Physik3“.

Man kann einen Fahrbahnwagen, auf welchem die Hallsonde befestigt ist, auf einer Führungsschiene senkrecht zur Spulenfläche bewegen. Mit dieser Hallsonde wird in Abhängigkeit vom jeweiligen Ort x eine Hallspannung U_H gemessen, welche proportional zu dem von der Spule erzeugten Magnetfeld ist. Diese Hallspannung wird mit PAKMA über den Messtyp **u_meter** erfasst. Der Ort des Gleiters wird mit einem Zählrad bestimmt und kann somit über den Messtyp **zählen** vom Rechner registriert werden.

Möchte man das Kernprogramm (s. S. 111) zum Versuch erstellen, so werden zu Beginn wiederum die Startwerte und Konstanten festgelegt. Im Anschluss daran werden für die beiden Messtypen die Vorbereitungen zum Messen getroffen. In unserem Beispiel sollen für die Messtypen **zählen** und **u_meter** die Messungen möglichst gleichzeitig erfolgen. Dies kann man realisieren, indem man für einen der Messtypen, z.B. **zählen**, eine zeitgesteuert intermittierende Messung und für den anderen Messtyp direkt im Anschluss daran eine Einzelmessung durchführt. Damit ergeben sich für das Kernprogramm folgende Anweisungen zur Vorbereitung der Messungen, welche für die beiden Messtypen nacheinander getroffen werden:

```
zählen;
vorb(1);           // Messschritt "vorbereiten" für zählen
vor_rück;
s_folg('i',dt);    // Festlegen der "Schrittfolge" für zählen
u_meter;
m_bereich(10,1);   // Festlegen des Meßbereichs
```

Für den Messtyp **u_meter** wird keine „Schrittfolge“ zum Messen festgelegt, damit bei Aufruf des Messbefehls innerhalb der Messschleife eine Einzelmessung durchgeführt wird.

Die Messungen erfolgen nun innerhalb einer REPEAT-UNTIL-Schleife, welche im Gegensatz zu den Beispielen 1 und 2 über den Ort x läuft, d.h., sobald der Messwert von x einen festgelegten Wert überschreitet, wird die Abarbeitung des Kernprogramms und damit die Messwertaufnahme beendet.

Bevor eine Messung durchgeführt werden kann, muss zunächst der entsprechende Messtyp aufgerufen werden. Man beginnt mit dem Messtyp **zählen**, da hierfür eine zeitgesteuert intermittierende Schrittfolge gewählt wurde. Bei Aufruf der Anweisung **mes** wartet das Programm nun bis das als Konstante festgelegte Zeitintervall dt abgelaufen ist, bevor eine Messung erfolgt.

Der Messwert kann dann über die Anweisung **mes_w** aufgerufen werden. Der seit der letzten Messung zurückgelegte Weg dx ergibt sich zu **dx:=mes_w*1_einh**. Hierbei gibt **1_einh** die Länge des Weges an, welcher zwischen zwei registrierten Impulsen bei Verwendung eines

Zählrades zurückgelegt wird und ist als Konstante zu Beginn des Kernprogramms zu definieren.

Den insgesamt zurückgelegten Weg erhält man durch die Anweisung:

```
x:=x+dx;
```

Anschließend soll die Hallspannung gemessen werden, wozu der Messtyp **u_meter** erneut aufzurufen ist. Da für diesen Messtyp eine Einzelmessung vorgesehen ist, wird der Messwert unmittelbar bei Aufruf der nun folgenden Anweisung **mes** erfasst. Mit dem Befehl **uh:=mes_w** wird dieser dann aufgerufen und einer Variable zugewiesen.

Nun folgt die Ausgabeanweisung des Kernprogramms, in welcher die Größen **x** und **UH** aufzuführen sind, da diese in einem Ausgaben-Fenster graphisch dargestellt werden sollen:

```
ausgabe(x,uh);
```

Zum Abschluss wird innerhalb der REPEAT-UNTIL-Schleife die Zeit **t** um das, zu Beginn des Kernprogramms als Konstante definierte, Zeitintervall **dt** erhöht:

```
t:=t+dt;
```

Dies ist für die zeitgesteuert intermittierende Messung des Ortes **x** erforderlich.

Die Messschleife kann nun erneut durchlaufen werden. Die Messwert-Erfassung wird beendet, sobald der Gleiter mit der Hallsonde einen bestimmten Ort **x** überschreitet.

Insgesamt hat das Kernprogramm folgenden Aufbau:

```
t:=0;x:=0;           // Anfangswerte
dt:=0.08;            // Konstanten
r0:=0.2;l_einh:=0.0009;
;
zählen;              // Messtyp-Aufruf
vorb(1);             // Messvorgaben für zählen
vor_rück;
s_folg('i',dt);
;
u_meter;             // Messtyp-Aufruf
m_bereich(10,1);
;
REPEAT               // Messschleifenbeginn
  zählen;            // Messtyp-Aufruf
  mes;               // Messung für zählen
  dx:=mes_w*l_einh;
  x:=x+dx;           // Berechnung des Ortes x
  ;
  u_meter;           // Messtyp-Aufruf
  mes;               // Messung für u_meter
  uh:=mes_w;
  ;
  ausgabe(x,uh);     // Ausgabeanweisung
  t:=t+dt;
UNTIL x>0.6;         // Abbruchbedingung der Messschleife
```

Abb. 4.7 zeigt die Darstellung eines möglichen Versuchsergebnisses.

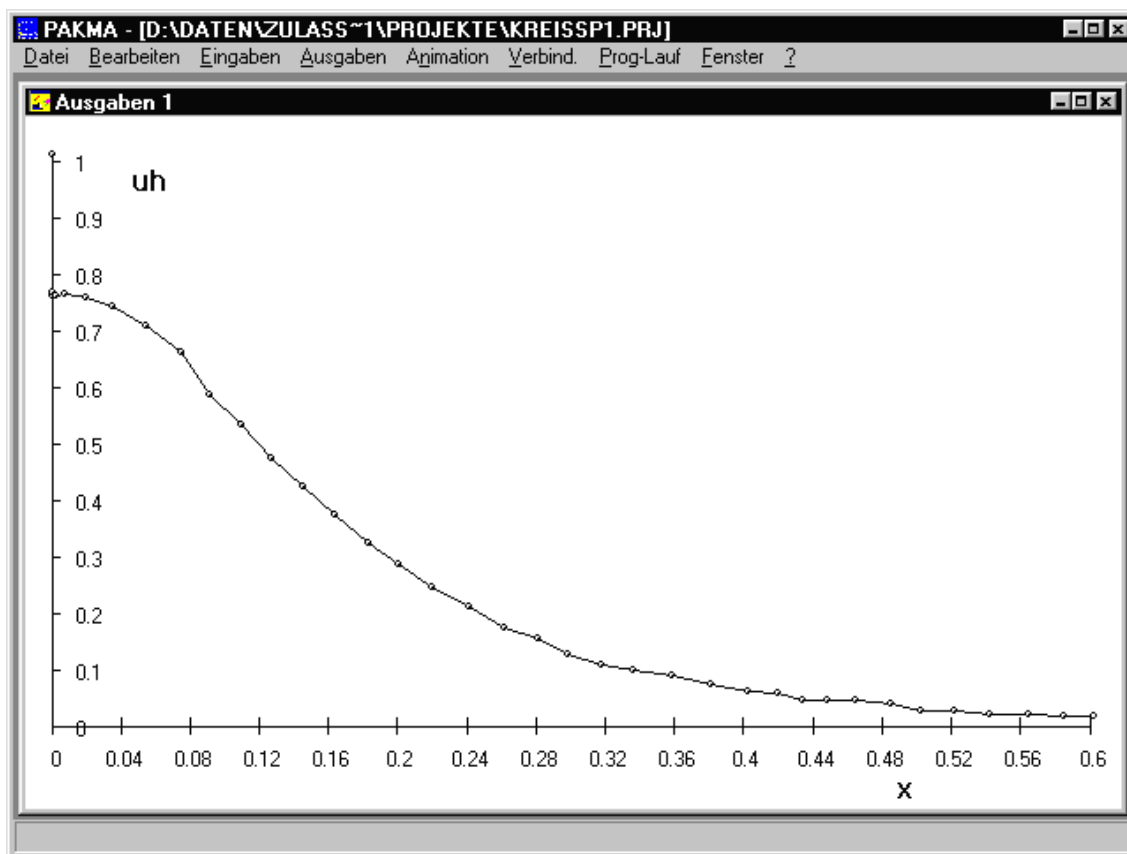


Abb. 4.7

Natürlich kann das Kernprogramm beliebig erweitert werden, indem z. B. eine Modellierung des physikalischen Sachverhaltes eingefügt wird.

4.5 Computermouse als zweidimensionaler Bewegungssensor

Da jede Computermouse eine zweidimensionale Bewegung erfasst, eignet sie sich grundsätzlich als Messwerkzeug zur Untersuchung solcher Bewegungen. Der schematische Aufbau (s. **Abb. 4.8**) macht die Grundlagen der zwei-dimensionalen Messwerterfassung deutlich.

Mit Hilfe zweier Lichtschranken werden durch die Bewegungen der zwei Lochräder elektrische Impulse erzeugt, die von Windows dazu verwendet werden, den Mauszeiger auf dem Bildschirm zu bewegen. PAKMA ist in der Lage, diese Informationen zu verarbeiten und als gemessene Größen im Ausgaben-Fenster darzustellen.

Hierzu ist jedoch ein spezieller Messbefehl im Kernprogramm nötig, der mitteilt, dass es sich bei der vorgenommenen Messung, um eine Messung mit der Maus handelt. Dieser neue Befehl lautet:

mauszählen

Er wird ebenso verwendet, wie die zuvor schon erwähnten Befehle **zählen** oder **u_meter**.

Im Folgenden wird ein Projekt vorgestellt, mit dem es möglich ist, Bewegungen mit der Maus aufzunehmen. Beim Arbeiten mit Projekten, bei denen die Maus als Bewegungssensor benutzt wird, ist darauf zu achten, dass die Geschwindigkeit der Maus auf den minimalen Wert eingestellt ist. Dies ist in der Systemsteuerung von Windows möglich. Den zugehörigen Dialog zeigt **Abb. 4.9**.

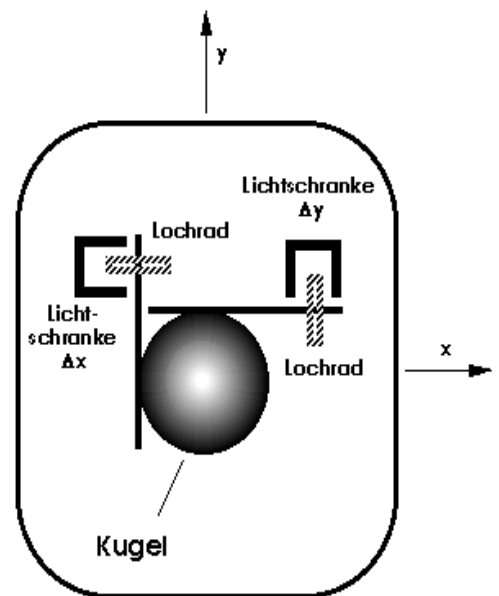


Abb. 4.8

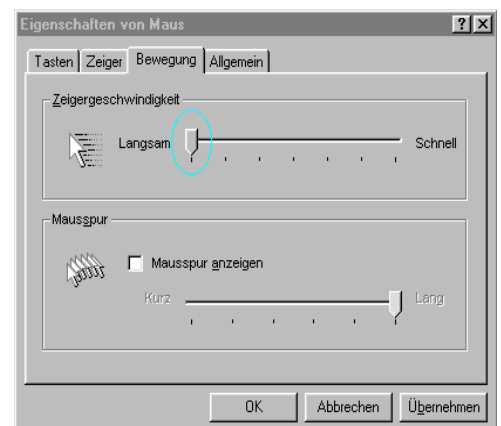


Abb. 4.9

Versuche zur 2-dimensionalen Bewegung

Mit einer Anordnung wie in **Abb. 4.10** können krummlinige Bewegungen, wie z.B. Kurvenfahrten nachvollzogen und analysiert werden. Als „Fahrbahn“ dient ein DIN A3-Blatt mit aufgedrucktem Koordinatensystem und Bahnkurve für Kreisbewegungen. Eine Einschränkung betrifft die "Fahrweise" mit der Maus. Damit die Zuordnung der Koordinaten zu den Sensorachsen der Maus erhalten bleibt, darf die Maus während der "Fahrt" nicht verdreht werden.

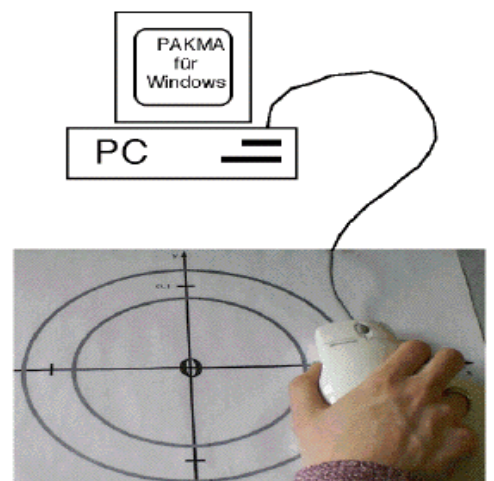


Abb. 4.10

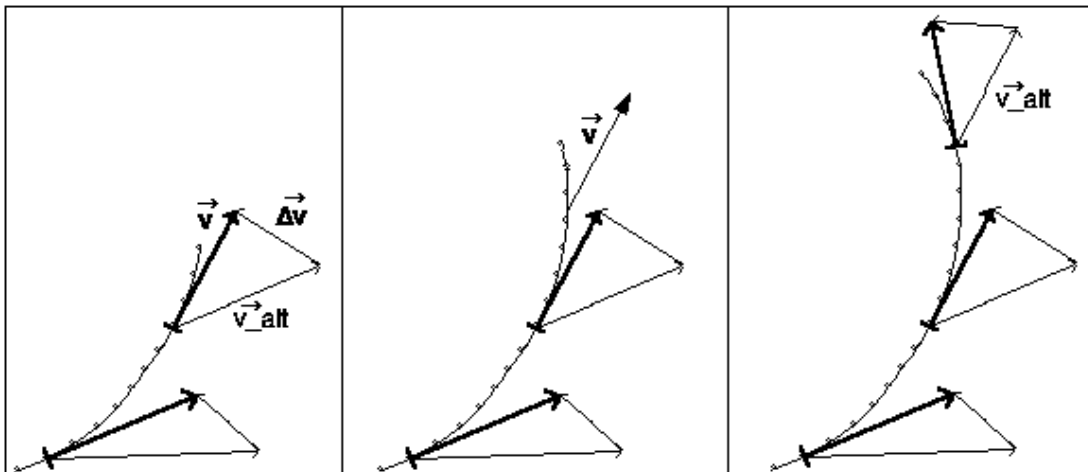


Abb. 4.11

Die Bildfolge in **Abb. 4.11** zeigt mehrere Phasen einer Kurvenfahrt mit Animation der Bestimmung des Geschwindigkeitsänderungsvektors. In **Abb. 4.12** sind für eine Kurvenfahrt mit der Maus die Bahnkurve mit Zeitmarken, sowie Geschwindigkeitsvektoren (einzelne Pfeile) und Beschleunigungsvektoren (Doppelpfeile) dargestellt. Damit lässt sich der handlungsorientierte Umgang mit der Beschleunigungsdefinition interaktiv umsetzen.

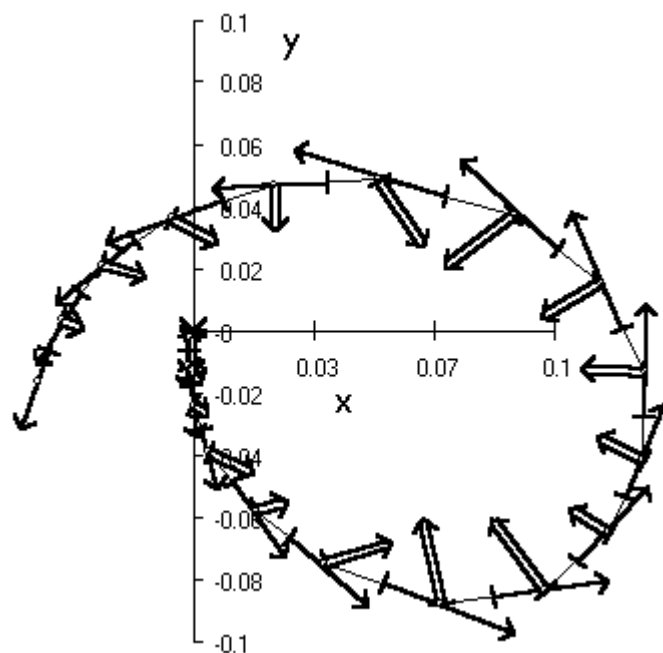


Abb. 4.12

4.6 Die Maus als eindimensionaler Low-Cost-Bewegungssensor

Während die Computermouse als 2-dimensionaler Bewegungssensor ohne jegliche Umbauten eingesetzt werden kann, ist es zur Registrierung und Analyse 1-dimensionaler Bewegungen nötig, die Maus zu verändern. Die Bewegung muss über einen Faden mit einer Sensorachse in der Maus gekoppelt werden. Dazu wird die Maus geöffnet, die Kugel entfernt und der Faden über die Achse einer Rasterscheibe geführt. Zur Verbesserung des Fadenlaufs und zur Vermeidung von Schlupf empfiehlt es sich, eine passende Kabeltülle s. Pfeil in **Abb. 4.13** oder einen Gummischlauch auf die Achse zu schieben und den Faden in der Tülle bzw. auf dem Schlauch laufen zu lassen. Außerdem muss der störende Einfluss von Umgebungslicht auf die Lichtschranke durch eine Abdeckung verhindert werden, s. **Abb. 4.13**

Grundsätzlich eignen sich sowohl PS/2- als auch serielle Mäuse zum Messen. Da bei der Untersuchung 1-dimensionaler Bewegungen die ins Experiment eingebaute Maus nicht mehr zum Bedienen des Computers zur Verfügung steht. Verwendet man idealerweise einen Computer mit einer PS/2-Maus zum Bedienen und einer zusätzlichen seriellen Maus zum Messen. Für beide Mäuse sind die im Kap. 4.5 aufgeführten Einstellungen vorzunehmen, bei manchen Notebooks mit PS/2-Touchpad kann die Einstellung für Touchpad und Sensormaus sogar getrennt erfolgen. Einziger Nachteil dieser Konfiguration ist, die Bedienmaus (Touchpad) darf während der Messung nicht benutzt werden.

Steht keine PS/2-Maus zur Verfügung, so ermöglicht der Umschalter in **Abb. 4.14** die wechselweise Aktivierung zweier serieller Mäuse zum Bedienen bzw. Messen. In diesem Fall muss die Umschaltung auf die jeweils „richtige“ Maus rechtzeitig von Hand erfolgen.

4.6.1 Versuchsbeispiel: Fahrbahnversuche

Im Gegensatz zu Lehrerdemonstrationsexperimenten kommt besonders bei Schülerversuchen der Low-Cost-Aspekt des Sensors Maus voll zum Tragen. Da Schülerfahrbahnen nicht so reibungsarm wie Luftkissenbahnen sind, verändert die zusätzliche Reibung des Sensors Bewegungsabläufe auch nur unwesentlich. Ein typisches Kernprogramm zum Messen, in dem s und v ermittelt wird, zeugt **Abb. 4.15**. Als Zeitintervall Δt sind Vielfache von 0,025 s zu wählen, angemessen sind Werte zwischen 0,05 und 0,2 s.

Das Ergebnis eines Beschleunigungsversuchs zeigt die **Abb. 4.16**. Durch leichtes Anheben der Fahrbahn wurde ein Reibungsausgleich vorgenommen, so dass damit sogar sehr gut Messexperimente zum 2. Newtonschen Axiom durchgeführt werden können.

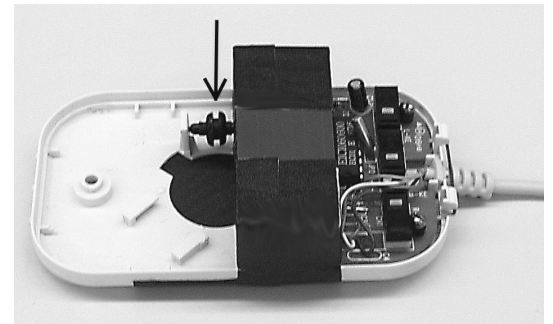


Abb. 4.13

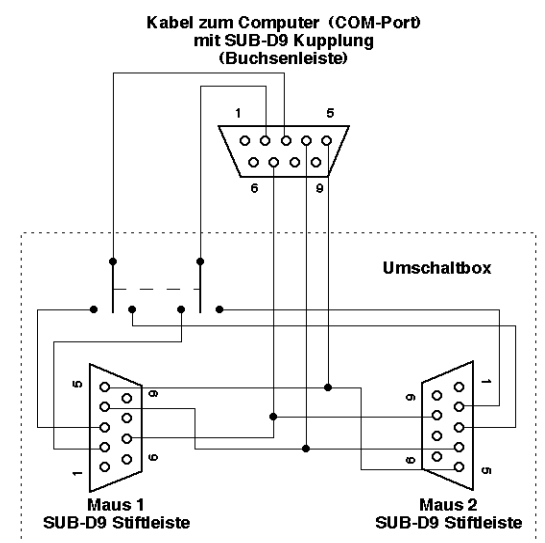


Abb. 4.14

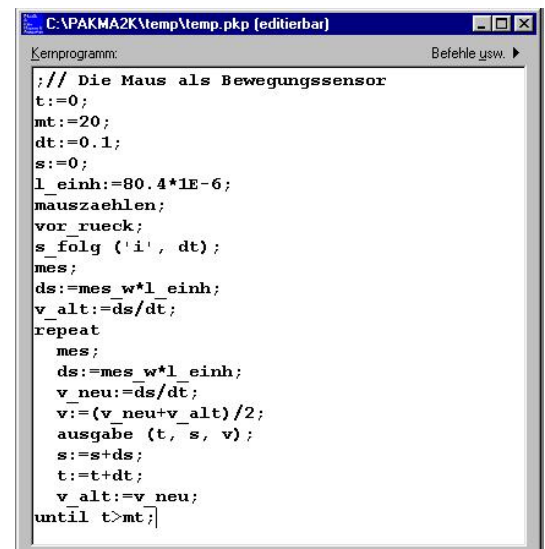


Abb. 4.15

Einen etwas anderen Aufbau mit einer Leybold Schülerfahrbahn, bei dem die Ankopplung der Bewegung an die Maus mit einem umlaufenden Faden realisiert ist, zeigt die **Abb. 4.17**. Die Maus ist mit einem geeignet gebogenen Alublech Stärke 1,5 mm, Breite ca. 3 cm auf einem Klemmreiter mit Einzel-Umlenkrolle so montiert, dass der Faden über die Umlenkrolle und die Kabeltülle läuft.

Typische Ergebnisse einer Fahrt auf der schiefen Ebene mit Reflexion an einer Feder am Ende der Fahrbahn sind in **Abb. 4.18** dokumentiert. Man sieht sehr gut die unterschiedlich große Beschleunigung bei der Tal- und Bergfahrt.

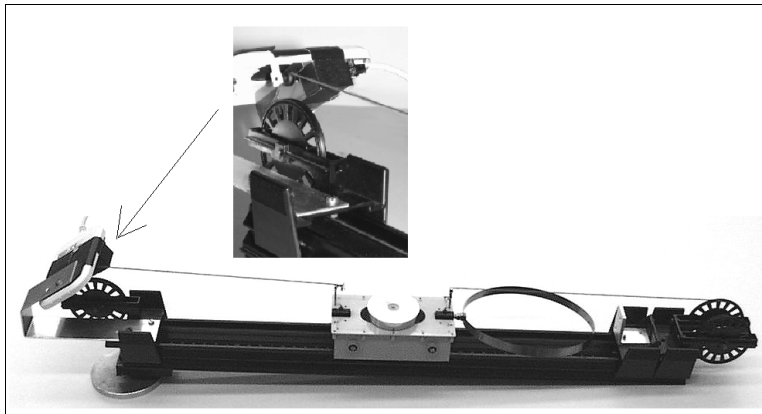


Abb. 4.17

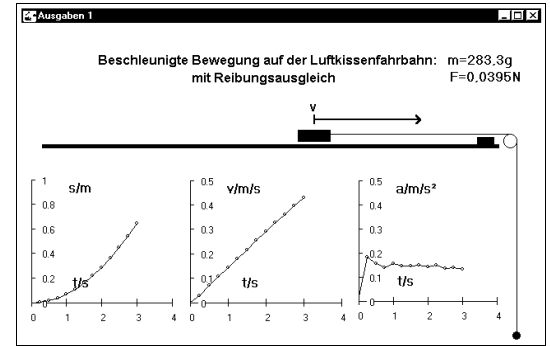


Abb. 4.16

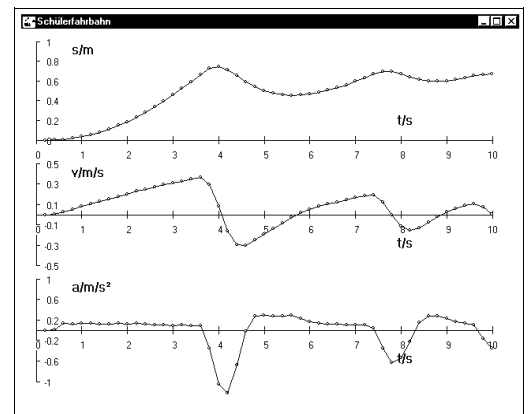


Abb. 4.18

4.6.2 Versuchsbeispiel: Federpendel

Auch für die Untersuchung der Bewegungsabläufe beim Federpendel eignet sich die Computeraus als Bewegungsmesswandler ausgezeichnet. Der Aufbau kann wie in **Abb. 4.19** erfolgen. Die Sensoranordnung wurde von der Schülerfahrbahn übernommen und der Faden durch das Innere der Schraubenfeder über Umlenk- und Mausrolle zu einem kleinen Spannungsgewicht von 2 bis 5 g geführt. Diese Fadenführung ermöglicht die Übertragung der Bewegung auf die Mausrolle bei relativ geringer Reibung. Ergebnisse sind in **Abb. 4.20** dargestellt.

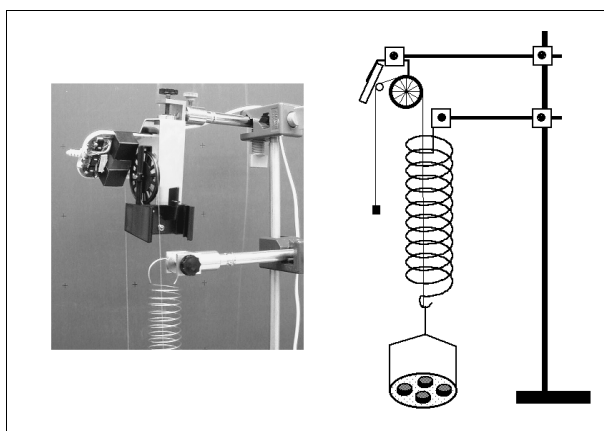


Abb. 4.19

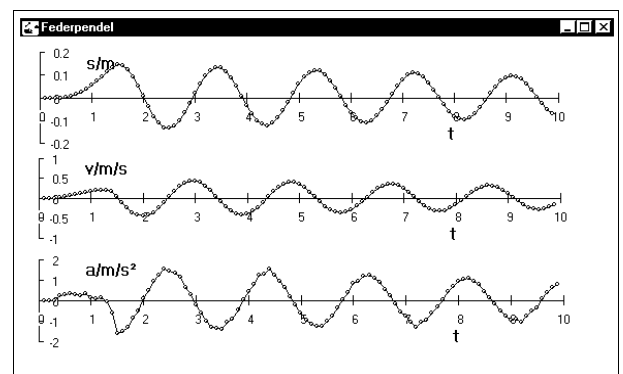


Abb. 4.20

4.7 Messen mit der seriellen Schnittstelle

Neu in PAKMA 2000 ist die Möglichkeit, mit der seriellen Schnittstelle Messungen durchzuführen. Die Messwerte werden direkt von den Messgeräten übernommen, die PAKMA-Interface-Box ist hierfür nicht nötig.

Zunächst müssen die verwendeten Geräte in PAKMA angemeldet werden. Im Menüpunkt „Eingaben“ kann unter „Serielle Geräte definieren...“ eine Liste der verwendeten Messgeräte angelegt werden. Mit „Neu“ wird ein neues Gerät angemeldet, dazu werden Geräte- und Schnittstelleneinstellungen mit dem Formular aus Abb. 4.21 abgefragt, die der Herstellerdokumentation zu entnehmen sind.



Abb. 4.21

Der Aufruf der Messgeräte im Kernprogramm erfolgt über den Befehl

Seriell(<comport>,'Gerätename')

wobei für <comport> die entsprechende Nummer z.B. 1 oder 2 einzusetzen und der Gerätename als String zu kennzeichnen ist, also z.B. 'Multimeter.' Um eine Messdatenablage zu wählen wird

Vorb_Seriell(<comport>,<Ablage>)

mit Ablagen= 1 bis 8 verwendet. Die Aufnahme eines Messwerts und die Zuweisung zu einer Kernprogrammvariable erfolgt über

mes_w_seriell(<comport>)

Die genaue Verwendung dieser Befehle ist dem Beispiel unten zu entnehmen. Der separate Befehl 'mes' entfällt hier, ebenso 's_folg()', da die Messintervalle nicht durch die Interface-Box vorgegeben werden.

Das Kernprogramm für die Messung einer Spannung mit dem Multimeter ME-32 am COMport1 lautet dann folgendermaßen:

```
Seriell(1,'Multimeter');      // An COM 1 angeschlossenes Multi-
                               // meter anmelden
Vorb_Seriell(1,3);            // Messdatenablage von COM1 in
                               // Messdatenablage Nr. 3
repeat
  U:=mes_w_seriell(1);        // Messwertabfrage und Zuweisung
  ausgabe(U);
  Schleifenzeit(0.5);        // Messwertablesung alle 0.5 s
until taste;
```

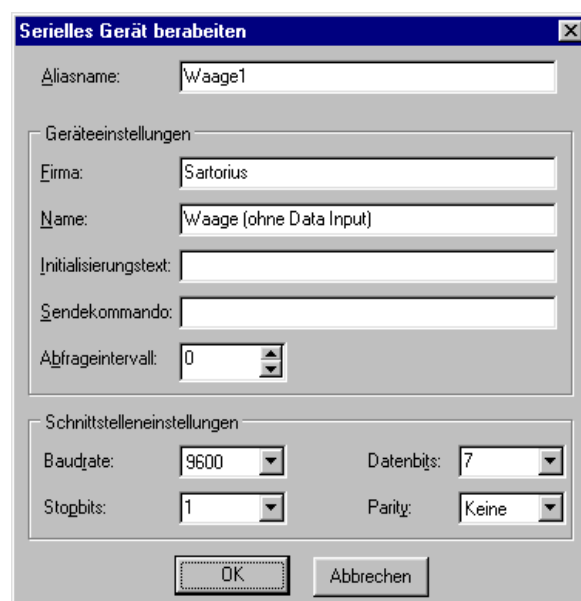


Abb. 4.22

Es können auch zwei Messgeräte gleichzeitig über die seriellen Schnittstellen abgefragt werden. Beide Geräte müssen dazu durch den Befehl **Seriell(<comport>,'Gerätename')** aufgerufen werden. Es ist darauf zu achten, beiden Geräten verschiedene Messdatenablagen zuzuweisen. Ein mögliches Beispiel ist die Messung der Kondensatorspannung und der Kraft der Kondensatorplatten aufeinander (vgl **Abb. 4.23**).

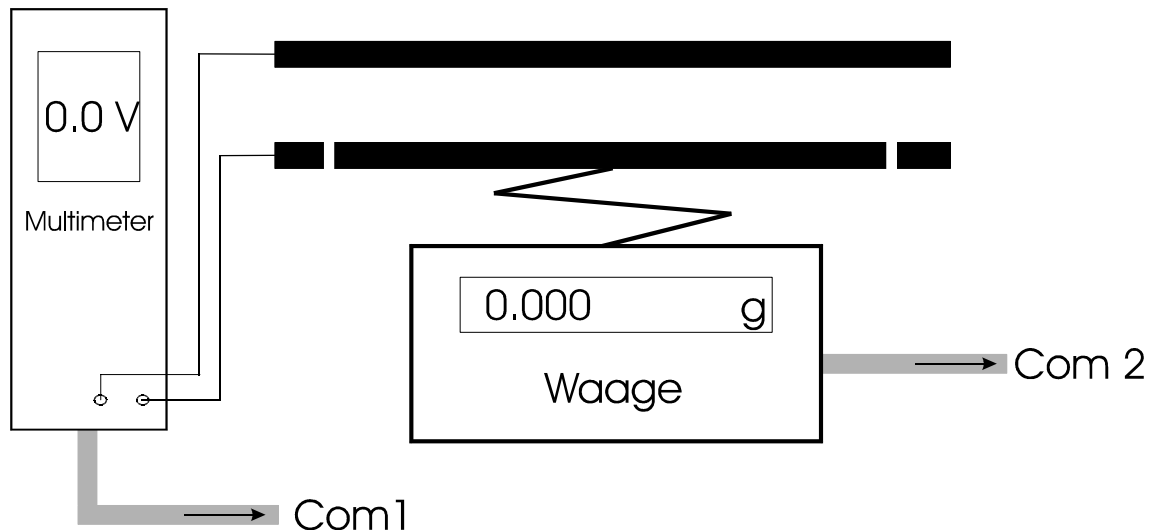


Abb. 4.23

Das zugehörige Kernprogramm lautet:

```

Seriell(1,'Multimeter');           //An COM1 angeschlossenes Multi-
                                   meter anmelden

Seriell(2,'Waage1');               //An COM2 angeschlossene Waage
                                   anmelden

Vorb_Seriell(1,3);                 //Messdatenablage von COM1 (Mul-
                                   timeter) in Messdatenablage Nr. 3

Vorb_Seriell(2,4);                 //Messdatenablage von COM2
                                   (Waage1) in Messdatenablage Nr. 4

repeat
  if start__>0 then                //Messung nur bei Betätigen der
                                   Ablauf-Taste
  begin
    U:=Mes_w_Seriell(1);           //Messdatenabfrage und Definition
                                   der Variablen U
    F:=Mes_w_Seriell(2);           //Messdatenabfrage und Definition
                                   der Variablen F
    U2:=U*U;                       //Definition der Variablen U2=U²
  end;
  Ausgabe(U,U2,F);
until taste;
```

Die Kraft F der Platten aufeinander wird im Ausgabefenster über der Kondensatorspannung U und über U^2 (Variable $U2$) aufgetragen. Der lineare Zusammenhang bei der Auftragung über U^2 ist in **Abb. 4.24** deutlich zu erkennen.

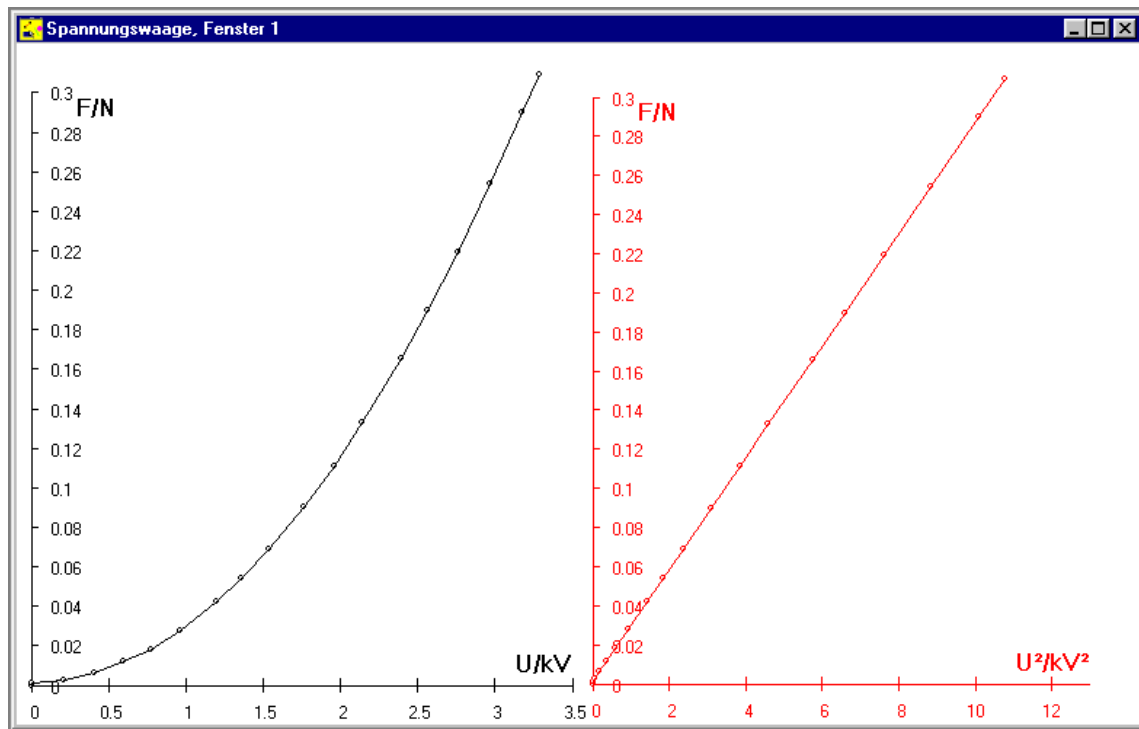


Abb. 4.24

Die obigen Graphen wurden bei einem Plattenabstand von 2mm aufgenommen. Die Messergebnisse für weitere Plattenabstände sind im beigefügten PAKMA Projekt u_waag.prj (siehe „Das elektrische Feld – Spannungswaage“) enthalten.

5 PAKMA-Befehle und -Funktionen

Im Editor des PAKMA 2000 eingegebene Texte werden in einem Parser so umgesetzt, dass sie dann von Turbopascal für Windows kompiliert werden können. Damit der Editor auch ohne PASCAL-Kenntnisse benutzt werden kann, ergänzt er manche Formelemente selbständig. Daher müssen bei PAKMA auch die Variablen nicht eigens deklariert werden, sondern werden, wenn sie benutzt werden, automatisch deklariert und mit 0 initialisiert. Zwischen Groß- und Kleinschrift wird nicht unterschieden.

5.1 Variablen

<code>var</code>	Realvariable. Dies sind die üblichen Variablen in PAKMA. Die Rechengenauigkeit beträgt 11 bis 12 Stellen
<code>var%</code>	Integervariable. Diese sind 16 Bit Ganzzahlen im Wertebereich von -32768 bis 32767.
<code>var\$</code>	Stringvariable. Diese Variablen dienen zur Verwaltung von Zeichenketten, können aber nicht mit ausgabe() übergeben werden.
<code>var [nr]</code>	Arrays werden definiert durch : <code>dim var[feldgr.]</code> Dabei werden <code>feldgr.+1</code> Felder angelegt (<code>var[0]</code> bis <code>var[feldgr.]</code>).

Alle Variablen, die im PAKMA-Kernprogramm benutzt werden, werden vor Programmstart mit 0 initialisiert, es sei denn sie erhalten im Startwerte-Menü einen anderen Wert.

5.2 Deklaration von Variablen

Wenn Variablen nicht durch eine Anweisung in der folgenden Art deklariert werden, werden sie automatisch als Real-Variablen bzw. – wenn sie auf ‘%’ oder ‘\$’ enden – als Integer- oder String- Variable deklariert.



```
var i, j : Integer;    Deklariert i und j als Integervariable
var r, x : Real;      Deklariert r und x als Realvariable
var s, t : String;    Deklariert s und t als Stringvariable
dim a[feldgr.];       Definiert a als Array von Realvariablen
```

Variablen, die durch eine derartige Anweisung deklariert worden sind, dürfen zur Verdeutlichung mit einem ‘?’ am Ende versehen werden.

Es existieren noch weitere Variablentypen, die man in PAKMA verwenden kann. Hierfür wird empfohlen, diese in einem PASCAL-Handbuch nachzuschlagen.



5.3 Schleifenbefehle und bedingte Ausführung von Anweisungen

Anweisungen in den folgenden Schleifen werden vom PAKMA-Editor automatisch eingerückt.

```
begin                                Anweisungen und Funktionen lassen sich mit begin und end zu
    [Anweisungen]                   Blöcken verbinden, so ist es möglich, mehrere Anweisungen in
end;                                 einer Schleife bzw. Abfrage unterzubringen.
```

```
for var%:=startw. to
(bzw. downto) endw. do
    [Anweisung]
```

Mit der **for**-Schleife lässt sich die Anweisung mehrfach hintereinander durchführen, wobei die Intervariable **var%** die Werte von **startw.** bis **endw.** in aufsteigender (bzw. absteigender (**downto**)) Reihenfolge annimmt. Sollen mehrere Anweisungen in der Schleife durchgeführt werden, müssen diese mit **begin** und **end;** geklammert werden.

Wird die Laufvariable nicht als Intervariable eingeführt, so ergänzt der Editor das **,%'** automatisch beim Verlassen der Zeile.

Wird eine for -Schleife in einer procedure oder function verwendet, muß die Laufvariable mit VAR deklariert werden.

```
repeat
    [Anweisung1]
    [Anweisung2]
until Bedingung;
```

Repeat-Schleife. Diese Schleife wird solange durchgeführt, bis die Bedingung hinter **until** *wahr* ist.

```
if Bedingung then
    [Anweisung1]
[else]
    [Anweisung2]
```

If-Abfrage. Ist die Bedingung *wahr*, wird die Anweisung1 ausgeführt, sonst wird, wenn vorhanden, die Anweisung2 ausgeführt.

Sollen mehrere Anweisungen in den einzelnen Fällen durchgeführt werden, müssen diese mit **begin** und **end;** geklammert werden.

```
case var% of
    wert_1:
        [Anweisung 1];
    wert_2,wert_3:
        [Anweisung 2]
    wert_4..wert_5:
        [Anweisung 4]
    wert_m:
        [Anweisung n]
[else]
    [Anweisung n+1]
end;
```

Nimmt die Intervariable **var%** einen der Werte **wert_1 ... wert_m** an, so wird die dahinterstehende Anweisung ausgeführt; sonst wird, wenn vorhanden, die Anweisung n+1 ausgeführt. Sollen mehrere Anweisungen in der Schleife durchgeführt werden, müssen diese mit **begin** und **end** geklammert werden. **var** darf z. B. auch die Zeichenvariable **taste** sein, jedoch keine String- oder Realvariable.

5.4 Rechenoperationen

+, -, *, /	Grundrechenarten
:=	Zuweisung (z. B. x:=x+1)
()	Klammern in Rechenausdrücken
sin(var)	Sinus (var : Winkel im Bogenmaß)
cos(var)	Kosinus (var : Winkel im Bogenmaß)
arctan(var)	Arkustangens (Ergebnis im Bogenmaß)

<code>pi</code>	ist mit maximaler Genauigkeit bereits vordefiniert
<code>ln (var)</code>	Logarithmus ($\text{var} > 0$)
<code>sqr (var)</code>	Quadrat
	Weitere Potenzen durch Multiplikation (z. B. $v^3 = v * v * v$) oder $v^n = \exp(n * \ln(v))$.
<code>sqrt (var)</code>	Quadratwurzel
<code>exp (var)</code>	Exponentialfunktion
<code>round (var)</code>	Rundet var auf nächsten Integerwert
<code>trunc (var)</code>	Schneidet bei var die Nachkommastellen ab
<code>abs (var)</code>	Liefert den Betrag von var
<code>random</code>	Erzeugt Zufallszahlen im Bereich von 0 bis 1 (0 inklusive, 1 exklusive)
<code>random(n)</code>	Erzeugt ganze Zufallszahlen im Bereich 0 bis n-1.

5.5 Logische Operationen

Um komplexere Bedingungen formulieren zu können, können einzelne Bedingungen mit folgenden logischen Operationen zu einer Bedingung verknüpft werden.

<code>((Bedingung1) and (Bedingung2))</code>	Bedingung erfüllt, wenn beide Bedingungen erfüllt sind
<code>((Bedingung1) or (Bedingung2))</code>	Bedingung erfüllt, wenn mind. eine der Bedingungen erfüllt ist

5.6 Sonstige Funktionen

<code>n\$:=taste</code>	Tastenabfrage bei laufendem Programm. Die Variable taste wird nur beim Aufruf des Befehls ausgabe neu gesetzt. Ggf. gedrückte Umschalttasten haben keinen Einfluß auf den Wert von taste . Mit taste können direkt nur alphanumerische Zeichen abgefragt werden. Um die übrigen Zeichen abzufragen, müssen die ASCII-Codes verwendet werden.
<code>// Kommentar</code>	Um ein Kernprogramm besser zu erklären, können Kommentare, d. h. Text, der keinen Einfluß auf den Programmablauf hat, verwendet werden. Jeder Text, der nach <code>//</code> bis zum Zeilenende oder zwischen <code>{</code> und <code>}</code> steht, wird als Kommentar angesehen.
<code>{ Kommentar }</code>	

5.7 Kernprogramm als `procedure kern`

Wenn mit selbstdefinierten Prozeduren, Funktionen und externen Prozeduren gearbeitet werden soll, ist es nötig, dass das Kernprogramm die Form einer **procedure kern** erhält, siehe das Beispiel in **Abschnitt 5.8.1**. Dann erfolgt auch keine automatische Initialisierung aller Variablen auf Null. Diese muß, genauso wie die Zuordnung der Startwerte zu den Variablen, durch die Aufrufe **nullwerte** und **startwerte** erfolgen.

5.8 Prozeduren und Funktionen

```
uses unitname;
```

Einbinden einer Pascal Unit.

bzw.

```
use unitname;
```

Dieser Befehl muß, wenn Units benutzt werden sollen, am Anfang des Programms vor **procedure kern**, stehen. Es ist dann auch notwendig, dass das Hauptprogramm die Form einer procedure mit Namen **kern** hat, s.o.

```
procedure dings
```

Wie in Pascal, ist es in PAKMA auch möglich Unter-routinen als (**procedure** und **funktion**) zu schreiben, die auch Übergabe von Parametern (z. B.: a,b,c :Real) erlauben. Funktionen geben auch einen Wert zurück; wenn nicht anders angegeben, ist der Rückgabewert eine Realvariable. Sollen Unter-routinen benutzt werden, muß das Hauptprogramm die Form einer **procedure** haben mit dem Namen **kern**.

```
procedure dings (a,b,c)
```

```
procedure dings (a,b,c :real)
```

```
function dings[(a,b,c)]
```

```
function
```

```
dings[(a,b,c:real)]:real
```

5.8.1 Beispiel

```
procedure kern
```

Anfang des PAKMA-Hauptprogramms

```
begin
```

```
  Nullwerte;
```

In diesem Fall notwendig, s. o.

```
  Startwerte;
```

In diesem Fall notwendig, s. o.

```
  [Anweisungen]
```

Beliebige Anweisungen

```
  b:=Doppelt(a);
```

Aufruf der **function Doppelt**

```
  [Anweisungen]
```

Weitere Anweisungen

```
  do_it;
```

Aufruf der **procedure do_it**.

```
end;
```

Ende des Hauptprogramms

```
funktion Doppelt(x: Real):Real
```

Deklaration der **function Doppelt**

```
begin
```

```
  return 2*x;
```

Übergabe des Rückgabewertes

```
end;
```

Ende der **function Doppelt**

```
procedure do_it
```

Deklaration der **procedure do_it**

```
begin
```

```
  [Anweisungen]
```

```
end;
```

Für genauere Informationen und weitergehende Möglichkeiten bei der Verwendung von Units, Prozeduren und Funktionen, auf ein Pascal- Handbuch verwiesen.

5.8.2 Abweichungen vom Pascal-Standard

Unterprogramme, die im Hauptprogramm verwendet werden, dürfen auch hinter dem Hauptprogramm stehen. Jedoch dürfen Unterprogramme, die vor dem Hauptprogramm stehen, nicht in Unterprogrammen hinter dem Hauptprogramm verwendet werden.

Variablen-, Konstanten- oder Typendeklarationen dürfen nicht länger als eine Zeile mit 128 Zeichen sein. Mehrere jeweils durch **VAR**, **CONST** oder **TYPE** eingeleitete Zeilen sind jedoch zulässig.

5.9 PAKMA-spezifische Befehle

5.9.1 Anweisungen zur Initialisierung bzw. Rücksetzen

Nullwerte	Setzt alle Variablen auf Null
Startwerte	Setzt alle Variablen für die Startwerte existieren auf diese Startwerte.

5.9.2 Messtypen

zählen	Zählen von Impulsen in dt
mauszählen	Zählen von PC-Maus Impulsen in dt
	Bei Verwendung von mauszählen wird automatisch 2-kanaliges Messen und vor_rück aktiviert.
u_meter	Spannung messen
bit_aus	Bitmuster ausgeben
bit_ein	Bitmuster abfragen

5.9.3 Einstellungen für alle Messtypen

vorb (abl)	Ablegen der Messwerte eines Durchgangs in der internen Ablage abl Es existieren 16 solche Ablagen. Werden allerdings Parallelmessungen in n-Kanälen durchgeführt, belegt ein Messdurchgang n Spalten, entsprechend muss dann $abl \leq 16/n$ sein. Wird diese Zahl überschritten, wird ein Fehler gemeldet.
s_folg (<i>mode</i> , dt)	Messmodus und Zeitschritt festlegen mode = i intermittierend (gleiches Intervall dt zwischen 2 Messungen) Die Messwerte werden sofort im Kernprogramm bearbeitet und gegebenenfalls ausgegeben (Echtzeit). Trifft das Kernprogramm vor Ablauf des Zeitintervalls dt auf einen Messaufruf, so wird noch so lange gewartet, bis seit der letzten Messung dt s vergangen sind. mode = f fortlaufend beim ersten Messaufruf werden alle Messungen durchgeführt und anschließend verarbeitet und ausgegeben. Der Vorteil gegenüber dem intermittierenden Messen ist, dass in kleineren Zeitschritten dt gemessen werden kann. Die Minimalwerte für dt hängen vom jeweiligen Rechner ab. mode = e Einzelmessung. Bei jedem Messaufruf wird eine Messung durchgeführt.
f_länge (n)	Nur wenn mehr als 1024 Messungen in einem Durchgang aufgezeichnet werden sollen, ist diese Anweisung erforderlich. Anzahl der dann zu speichernden Messwerte ist (n*1024) mit n=1,2,3,4

5.9.4 Einstellungen für zählen

vor_rück	Messen in beide Richtungen (ohne diesen Aufruf : nur in eine Richtung)
sonar (i)	Statt Zählrad Sonarmessung auf i Kanälen durchführen (i = 1 oder 2). Die übrigen Kanäle bleiben Zählkanäle.

5.9.5 Einstellungen für mauszählen

mausunsichtbar Schaltet den Mauszeiger während der Messung aus.

5.9.6 Einstellungen für u_meter

m_bereich(u_max, kanal) Stellt den Messverstärker für den Kanal **kanal** auf die angegebenen Messbereich **u_max** ein. Vorangestellt ist u_max = 10 Volt für alle Kanäle. Mit kanal=0 werden alle Kanäle auf den angegebenen Wert eingestellt.

Folgende Messbereiche sind möglich : u_max = 10V; 1V; 0.1V; 0.01V.

mes_warte(mms) Erhöht die Wartezeit zum Einschwingen des Verstärkers (größeres mms => größere Zeit). Nur nötig, wenn u_max ≤ 1V. Die Werte für t werden in Microsekunden angegeben. Üblich sind Werte zwischen 10 und 1000 für 486'er bzw. Pentium Rechner

n_auf1 Stellt den 8-Bit Modus ein. Gilt nur für die Spannungsmessung mit alten C64-Modulen.

Hier gelten auch für m_bereich andere Grenzen und zwar :

5.9.7 Befehle zum Messen

mes Durchführen einer einkanaligen Messung mit dem aktuell aufgerufenen Mess-typ

Ausnahme : Bei **mauszählen** wird eine 2-Kanal Messung durchgeführt

mes_p (n) Durchführen einer Parallelmessung auf n Kanälen

mes_r (n) Durchführen einer einkanaligen Messung im Kanal n. Dies sollte nur im s_folg-Modus 'e' (Einzelmessung) verwendet werden. Der Befehl kann sinnvoll sein, um für verschiedene Zählkanäle unterschiedlich lange Meßzeiten vorzugeben.

5.9.8 Übergabe der Messwerte an Variablen

v := mes_w Übergabe des Messwerte der letzten Messung an die Variable v

v := mes_wp (n) Übergabe des Messwerte des n-ten Kanals der letzten Parallelmessung an die Variable v

v := sp_w(r, i) Übergabe des Messwertes der r-ten Messung des i-ten Kanals der letzten Parallelmessung an die Variable v.

Wurde **vorb**(k) benutzt muß man statt i, n*k+i benutzen, wobei n die Anzahl der Kanäle ist.

5.9.9 Einlesen von Eingabewerten während des Programmlaufes

y:=schieber (x) Während des Programmlaufs können über das Eingabeelement: „Schieber x“ Werte eingelesen und einer Variablen y zugeordnet werden.

Hinweis: Vor dem Programstart muß das Ausgabefenster aktiviert sein, in dem der zu verändernde Schieber liegt. Zum Einlesen muß über Eingaben/Schieber ein Schieberelement im Ausgabefenster erzeugt werden.

y:=schalter (x)	Entsprechend schieber(x); das Schalterelement kann hier jedoch nur zwei Werte annehmen.
y:=taster (x)	Entsprechend schalter(x); das Schalterelement wechselt hier zwischen zwei Werten.

5.9.10 Bereitstellen von Variablen für die Ausgabeelemente

ausgabe(var1,var2,...) Bereitstellen der Variablen var1,var2,.. für Ausgabeelemente. Es können bis zu 32 Variablen übergeben werden.

5.9.11 Warte-Routine

warte_sek (t)	Der Rechner wartet an dieser Anweisung t Sekunden in Vielfachen von 0,01s, unabhängig von der Taktfrequenz des Rechners, bevor er zur nächsten Anweisung weitergeht.
init_schleifenzeit	Initialisiert die Verwendung von schleifenzeit.
schleifenzeit (t)	Der Rechner wartet an dieser Anweisung t Sekunden seit dem letzten Aufruf von schleifenzeit oder init_schleifenzeit.

5.9.12 Befehle zum Steuern über die Interfacebox

relais (a)	Schaltet das Relais (a=1 : ein und a=0 :aus) Grundeinstellung ohne Aufruf : ausgeschaltetes Relais. Bei Programmabbruch oder -ende wird das Relais ausgeschaltet
ausg (bits)	Gibt den Wert von bits als Bitkombination aus. Voraussetzung hierfür ist, dass vorher der Meßtyp bit_aus gewählt wurde.

5.9.13 Funktionen zum Abfragen des Programm-Modus

m=reprowert	Gibt den Modus an, in dem das Kernprogramm zuletzt compiliert wurde : m = 0 : Original m = 1 : Modell m = 2 : Reproduktion m = 3 : wird bei Original und Reproduktion ausgegeben, falls im Rechner keine PAKMA-Interfacekarte eingebaut ist.
--------------------	--

5.9.14 Zeitverzögerung

warte_sek(t) ; Wartet für t Sekunden ($t \geq 0.01$).

5.9.15 Graphensteuerung

graf_löschen ;	Löscht alle Graphen in allen Graphendarstellungen
neu_graf	Unterbricht die Verbindungslinien aller Graphen für den nächsten Aufruf des Ausgabebefehls.

5.9.16 Neue externe Funktionen und Prozeduren in der Unit Pak_math

y:=sign (x)	Liefert in y das Vorzeichen von x, d. h. -1 wenn x negativ, 0 wenn x Null und 1, wenn x positiv ist.
--------------------	--

Pak_math

mittelung (x,t,n,x_m,t_m)	Mittelt über jeweils n Werte der Real-Variablen x und gibt den Zeitpunkt t_m aus. x_m ist der Mittelwert.
lin_naeherung (y,x,x_min,x_max)	Bildet die lineare Näherung von y zu x im Intervall [x_min, x_max] mit der Methode des kleinsten Quadrates.
lin_faktoren (y,a,b)	Die Real-Variablen a und b werden auf die Koeffizienten der linearen Näherung (s.o.) gesetzt.
y_n:= lin_kurve (y,x)	Setzt y_n auf die Werte der linearen Näherungskurve
quad_naeherung (y,x,x_min,x_max)	Bildet die quadratische Näherung von y zu x im Intervall [x_min, x_max] mit der Methode des kleinsten Quadrates.
quad_faktoren (y,a,b,c)	Die Real-Variablen a, b und c werden auf die Koeffizienten der quadratischen Näherung (s.o.) gesetzt.
y_n:= quad_kurve (y,x)	Setzt y_n auf die Werte der quadratischen Näherungskurve.
diffq_op (x,t,v,t_v);	Bildet den Differenzquotienten $v=dx/dt$ zum Zeitpunkt t_v.
y:= integral (x,t);	Berechnet das Integral von x über t.
x_alt:= alt_wert (x);	Gibt den vorherigen Wert von x.
x:= sum (dx);	Erhöht x um dx.
dx:= diff (x);	Weist dx den Wert der Änderung der Variablen x zu.

6 Anhang

In diesem abschließenden Kapitel werden einige Punkte aufgeführt, die helfen können, falls es zu Problemen beim Einsatz von PAKMA kommt. Zusätzlich finden sich am Ende wichtige Adressen, die Informationen zu Neuerungen bzw. Erweiterungen von PAKMA anbieten.

6.1 Fehlermeldungen

Im Folgenden werden einige Fehlermeldungen aufgelistet, die von PAKMA ausgegeben werden. Insbesondere können dies Meldungen auftreten, wenn selbst Projekte erstellt werden. Probleme die durch eine fehlerhafte oder beschädigte Installation auftreten, werden in **Abchnitt 6.2** behandelt.

Die Meldung, die in **Abb. 6.1** gezeigt wird, wird von PAKMA dann ausgegeben, falls im Kernprogramm unzulässige Eingaben gemacht wurden, die dann beim Compilieren entdeckt werden.

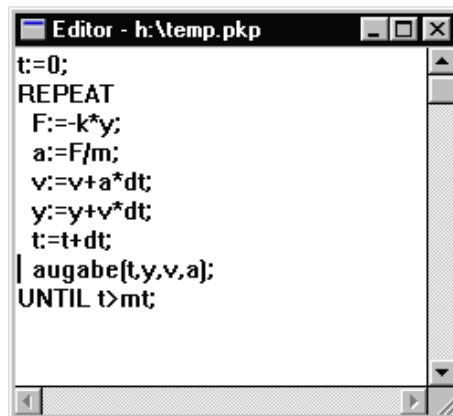


Abb. 6.2



Abb. 6.1

Nach Betätigen des Buttons **OK** befindet sich der Cursor zu Beginn derjenigen Zeile, in der der Fehler auftaucht. In diesem Fall wurde der Befehl **ausgabe** nicht richtig geschrieben, was in **Abb. 6.2** gezeigt wird.

Bei fehlerhafter Deklaration der Startwerte wird die Meldung aus **Abb. 6.3** ausgegeben. In diesem Fall werden einer Variablen im Startwertefenster keinerlei Werte zugewiesen (s. **Abb. 6.4**). Dies ist in PAKMA nicht erlaubt.

Variable:	Bezeichnung:	Vorgabe:	Letzter W.:	Neuer W.:
dt	Zeitintervall			
mt	Meßzeit	10	10	10
m	Masse in kg	1	1	1
k	Federkonst N/m	1	1	1
y	Anfangs-Auslenkung	2	2	2
v	Anfangsgeschwindigkeit	0	0	0

Abb. 6.4



Abb. 6.3

Variablen, die im Startwertefenster nicht auftauchen und die auch keine Wertzuweisung im Kernprogramm erhalten, bekommen automatisch den Wert Null. D. h. es erscheint dann keine Fehlermeldung.



Werden bei der Definition von Variablen (s. Abb. 6.6) Variablen benutzt, die nicht existieren, wird folgende Fehlermeldung ausgegeben. (s. Abb. 6.5), auch wenn die Bereichsgrenzen korrekt zugewiesen wurden (s. Abb. 6.7).

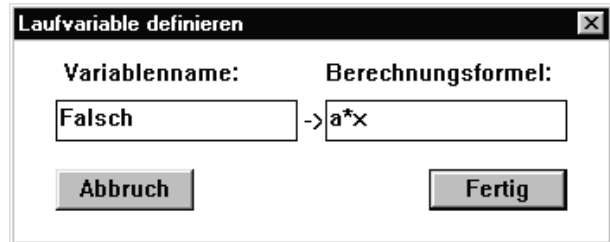


Abb. 6.6



Abb. 6.5

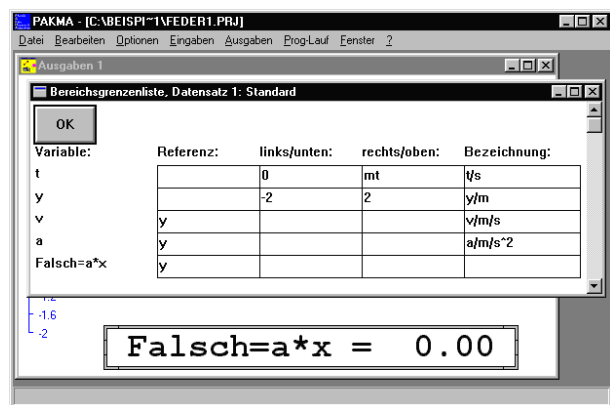


Abb. 6.7

Um diesen Fehler zu beseitigen, muss die ‚falsche‘ Variable gelöscht werden, bevor sie dann richtig definiert werden kann. Hierzu ist der Unterpunkt **Variable löschen** aufzurufen. Dieser Befehl muss dann noch bestätigt werden (s. Abb. 6.8).

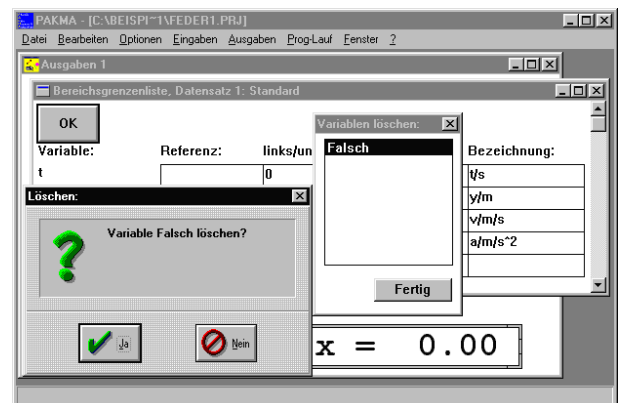


Abb. 6.8

6.2 Installations- und Laufprobleme

Treten prinzipiell beim Umgang mit PAKMA Probleme auf, so kann dies an einer fehlerhaften Installation liegen. Um dies überprüfen zu können, werden nun Dateien und Verzeichnisse aufgeführt, die sich auf der jeweiligen Festplatte befinden müssen.

Verzeichnis PAKMA

pakma.ini im Verzeichnis Windows

PAKMA Ordner im Unterpunkt Programme des Startmenüs.

Unterverzeichnis Temp im Verzeichnis PAKMA

pakma.exe im Verzeichnis PAKMA

pakmess.dll ebenfalls im Verzeichnis PAKMA und nicht im Verzeichnis Windows!

6.2.1 Einstellungen vornehmen

Die von PAKMA benötigten Informationen befinden sich in der Datei **pakma.ini**. Diese sollte bei einer Installation von PAKMA auf **C:** folgende Einträge haben:

[PAKMA]

PakmaDir=C:\PAKMA

TempDir=C:\PAKMA\TEMP

Wartezeit=2000

Um mit etwas älteren Rechnern zügiger arbeiten zu können, ist es ratsam, den Pfad für temporäre Dateien so abzuändern, dass dieser auf ein **RAMDRIVE** verweist. Dies wird unter Windows 95 so erzeugt, indem der Datei **config.sys** folgender Eintrag hinzugefügt wird:

DEVICE=C:\WINDOWS\RAMDRIVE.SYS 640 /E

Darauf wird bei nächsten Systemstart Festplattenplatz im Arbeitsspeicher simuliert. Dieser kann dann für temporäre Auslagerungen von PAKMA benutzt werden.

Befindet sich auf dem verwendeten Rechner nur eine Partition, nämlich **C:**, so lautet der entsprechende Eintrag in der Datei **pakma.ini** wie folgt:

TempDir=D:

6.2.2 Bekannte Laufprobleme

PAKMA läuft nicht von komprimierten Partitionen der Festplatte

PAKMA läuft nicht unter Windows NT

6.3 Bezugsquellen

Hardware zum Messen, wie z. B. Steckkarte, Interface-Box I und Sensoren sind über folgende Adresse erhältlich:

Firma
Microsystems
Neufahrner Str. 21
85748 Garching

6.4 Rückmeldungen / „Hotline“

Anfragen, „Fehlermeldungen“ und Wünsche sind an folgende Adresse zu richten:

Prof. Dr. D. Heuer
Lehrstuhl für Didaktik der Physik
Physikalisches Institut
Am Hubland
97074 Würzburg
Tel.: 0931/888-5787
Fax.: 0931/70 62 97

6.5 PAKMA im Internet

Die eMail- Adresse des PAKMA- Teams lautet:

pakma@didaktik.physik.uni-wuerzburg.de

Desweiteren besteht auch eine Zugriffsmöglichkeit über WWW auf PAKMA- News sowie Downloads von Aktualisierungen der Software, Projekten mit deren Dokumentation und diesem Handbuch.

Die Homepage des Lehrstuhls für Didaktik der Physik am Physikalischen Institut der Universität Würzburg lautet:

<http://www.didaktik.physik.uni-wuerzburg.de>

6.6 Literatur zu PAKMA

Beim Erstellen dieses Handbuches wurde folgende Literatur verwendet:

D. Heuer: Dynamische Physik-Repräsentation in Realexperimenten. Tagungsband DPG, 1993.

D. Heuer: Ladungsvorgänge am Kondensator. PdN-Ph, Jg. 43, Heft 3, S. 4 ff., 1994.

W. Reusch, Th. Grimmer, D. Heuer: Magnetfeld längs der Achse von Kreisströmen. Pdn-Ph, Jg. 43, Heft 3, S. 26 ff., 1994.

D. Heuer: Kurzanleitung zur Programmierumgebung PAKMA für Windows. (vom 19.12. 1994).

D. Heuer, R. Treffer: Handbuch: Physik erfahren durch Computer-Experimente, Bd. 2: Computer-Versuchsanalyse. Würzburg 1988 .

D.Heuer: Bewegungen "haargenau" messen mit Sonarmeter oder Laufrad. PdN-Ph 41, Heft 4, S. 4-8, 1992.

Stichwortverzeichnis

A

Animationselemente

- Benutzerdefiniertes Bitmaps 40
- Bitmap 40
- Breitpfeil 39
- Feder 40
- Rechteckkette 45
- Reihenvektor 42
- Text 41
- Uhr 41

Aufbau eines Kernprogramms 22

Ausgaben

- Autoskalierung 33
- Balkendarstellung 31
- Eigenschaften des Feldes wieder anzeigen 30
- Feld deaktivieren 30
- Feld erneut aktivieren 30
- Feld positionieren 29
- Graphenauswahl 33
- Graphendarstellung 32
- Graphenliste 32
- Großanzeige 28
- Größe des Feldes verändern 29
- Zeigerinstrument 30

Ausschneiden 10

B

- Bearbeiten 10
- Befehle und Funktionen 79
- Bereichsgrenzen 10, 14, 17, 24
- Bezeichnung 24
- Bildzwischenspeicher 11

D

- Das Kernprogramm 18
- Das Startwerte-Fenster 23
- Datei öffnen 10
- Datensatz umbenennen 26
- Datensatz wechseln 26
- dynamische Repräsentation 18

E

- Editieren 11
- Einfügen 10
- Eingaben
 - Bereichsgrenzen 10
 - Programmeditor 10

Startwerte 10

Variablen definieren 10

Experimental- Projekt 18

Externes Kernprogramm übernehmen 10

F

- Fehlermeldungen 87
- Funktionen 81

G

graphischer Editor 57

H

Hintergrundbild 11

I

Interface-Box 62

K

Kopieren 10

L

Letzter W 24

Logische Operationen 81

M

Menüpunkte

- Ausgaben 10
- Bearbeiten 10
- Datei 9
- Eingaben 10
- Fenster 12
- Prog.-Lauf 12
- Messbefehle 62
- Messen 62
- Modell 13, 18
- Modul laden 9
- Modul speichern 9

N

- Neuer W 24
- Neu Zeichnen 10

O

Original 13

P

- PAKMA Anweisungen 27
- PAKMA-Spezifische Befehle 83
- physikalisches Modell 48
- Programmeditor 10, 14f
- Projektbeschreibungen 10
- Projekte starten 13

Projekt-Info starten 10
Projekt öffnen 12
Prozeduren 81

R

Rechenoperationen 80
Reproduktion 13

S

Schieber/Schalter 10
Schleifenbefehle 79
Speichern 10
Startwerte 10, 14
Startwerte-Fenster 23
Stempeln 47
Systemvoraussetzungen 7

V

Variable 24
Variable definieren 16
Variablen 79
Variablen Autoskalieren 25
Variablen definieren 10, 14
Variablenliste bereinigen 25
Videoclip starten 10
Vielfachanimationselemente 41
Visedit 57
Visedit laden 10
Vorgabe 24

W

Wirkungsgefüge 57