



---

# Numerische Methoden zur Simulation verfahrenstechnischer Prozesse

Herrn Dr. rer. nat. Theo Ankel zum 65. Geburtstag

Erwin Dieterich, Gheorge Sorescu  
und Gerhart Eigenberger\*

---

In diesem Beitrag werden wesentliche Grundlagen und der derzeitige Stand der numerischen Berechnung des stationären und dynamischen Verhaltens verfahrenstechnischer Prozesse referiert. Durch Diskretisierung der örtlichen Ableitungen in den Modellgleichungen läßt sich das Gesamtmodell in ein System von Differentialgleichungen in der Zeit und von algebraischen Gleichungen, ein sog. DA-System, umformen. Die notwendigen Teilschritte zur Lösung des DA-Systems werden besprochen, auf bewährte Standardsoftware für diese Teilschritte sowie für die Lösung des gesamten DA-Systems wird hingewiesen.

**Numerical Methods for Simulation of Chemical Engineering Processes.** Essential fundamentals and the current state of the art in simulating the dynamic and the steady state behaviour of chemical engineering processes are discussed. It is shown that discretization of the spatial derivatives in the balance equations leads to a system of so-called DAE (differential algebraic equations), consisting of ordinary differential equations in time and algebraic equations. The paper discusses necessary steps to solve the DAE and mentions approved standard software for these steps as well as for the solution of the DAE as a whole.

---

## 1 Einführung

Verfahrenstechnische Forschung und Entwicklung basiert auf der Analyse und der gezielten Nutzung der maßgebenden physikalisch-chemischen Wechselwirkungen. Dabei gewinnen die mathematische Modellbildung dieser Wech-

selwirkungen und ihre numerische Simulation immer größere Bedeutung. Sie erlauben es, experimentelle Untersu-

---

\* Dipl.-Ing. E. E. Dieterich, Dr.-Ing. G. Sorescu und Prof. Dr.-Ing. G. Eigenberger, Institut für Chemische Verfahrenstechnik, Universität Stuttgart, Böblinger Str. 72, 7000 Stuttgart 1.

chungen gezielt auf abgrenzbare Teilaspekte zu konzentrieren und im Wechselspiel von Experiment und numerischer Simulation komplexe Zusammenhänge und Kopplungen aufzudecken und technisch zu nutzen.

Während die experimentelle Analyse eine lange und bewährte verfahrenstechnische Tradition besitzt, sind detaillierte mathematische Modellbildung und numerische Simulation Techniken, die bislang nur lose und in von Ort zu Ort sehr unterschiedlicher Intensität im Rahmen der verfahrenstechnischen Ausbildung vermittelt werden. Hinzu kommt, daß die Weiterentwicklung von Methoden und Verfahren der Rechnersimulation so schnell fortschreitet, daß es zunehmend schwieriger wird, den Überblick zu behalten.

Im vorliegenden Beitrag sollen daher zum einen die wesentlichen Grundlagen der numerischen Simulation skizziert und Standardmethoden herausgestellt sowie einige für ihre Vertiefung geeignete Quellen genannt werden. Zum anderen wird auf Entwicklungen eingegangen, die die Anwendung der numerischen Simulation beeinflussen und für ihren künftigen Einsatz bedeutsam werden können.

Im Gegensatz zum Informatiker oder (numerischen) Mathematiker sieht der Ingenieur in der Regel in der numerischen Simulation ein Handwerkszeug, mit dessen innerem Aufbau er sich nur in dem Maße beschäftigt, wie es zum zweckmäßigen Gebrauch unumgänglich ist. Auf dieser Prämisse basiert auch der vorliegende Beitrag. Allerdings zeigt es sich, daß man mit numerischen Standardwerkzeugen auch nur Standardprobleme der Rechnersimulation lösen kann und daß jeder Vorstoß ins Neuland zu einer Auseinandersetzung mit den in Frage kommenden numerischen Methoden zwingt. Auch für diese Aufgabe möchte der Beitrag Hinweise liefern.

Physikalisch begründete mathematische Modelle verfahrenstechnischer Apparate und Anlagen beruhen auf den Erhaltungsgleichungen von Masse, Impuls und Energie. Für die meisten Anwendungen werden sie bisher auf eine oder auf zwei Ortskoordinaten beschränkt; häufig wird nur der stationäre Zustand betrachtet. Die letztgenannte Einschränkung sei aber im folgenden fallengelassen, einmal, weil die Betrachtung der Dynamik für das Verständnis des verfahrenstechnischen Apparates oder Prozesses immer wichtiger wird, und zum anderen, weil die Mitnahme der Dynamik im Vergleich zur Berechnung des stationären Zustands häufig nur einen geringen Mehraufwand erfordert und gelegentlich sogar eine sicherere Konvergenz der Lösung liefert als eine rein stationäre Berechnung. Vor allem läßt sich die dynamische Simulation aber nach einem weitgehend einheitlichen Schema, der sog. „method of lines“, durchführen.

## 1.1 Einführungsbeispiel

Als Einführungsbeispiel sei das Modell eines isothermen Rohrreaktors betrachtet, das für eine maßgebende Reaktion durch die Materialbilanz für die Konzentration  $c$  einer Schlüsselkomponente beschrieben werden kann. Unter den Annahmen des örtlich eindimensionalen Diffusionsmodells lautet die Materialbilanz:<sup>1)</sup>

$$B \frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial z^2} - v \frac{\partial c}{\partial z} + S(c, z, t) \quad (1)$$

mit der Anfangsbedingung

1) Eine Zusammenstellung der Formelzeichen befindet sich am Schluß des Beitrags.

$$c(t = 0, z) = c^0(z) \quad (2)$$

und den Randbedingungen

$$D \left. \frac{\partial c}{\partial z} \right|_{z=0} = v(c(t, z=0) - c^*) \quad , \quad (3)$$

$$\left. \frac{\partial c}{\partial z} \right|_{z=L} = 0 \quad . \quad (4)$$

Der Quellterm  $S(c, z, t)$  beschreibt die meist nichtlinear von der Konzentration  $c$  abhängige Reaktionsgeschwindigkeit, die auch explizit von Ort und Zeit abhängen kann. Der in Gl. (1) enthaltene Kapazitätskoeffizient  $B$  besitzt den Wert 1. Seine explizite Mitnahme erweist sich aber im folgenden als sinnvoll.

Beziehung (1) stellt eine nichtlineare partielle Differentialgleichung (PDGL) vom Diffusions-Konvektionstyp dar und ist typisch für die Beschreibung örtlich eindimensional durchströmter verfahrenstechnischer Apparate.

Die übliche Methode zu ihrer numerischen Lösung ist die bereits angesprochene „method of lines“. Sie beruht auf der Diskretisierung des örtlichen Differentialquotienten

$$D \frac{\partial^2 c}{\partial z^2} - v \frac{\partial c}{\partial z} \quad (5)$$

zum Beispiel durch eine einfache äquidistante Differenzenapproximation an der Stelle  $l$  mit der Ortsschrittweite  $\Delta z$ :

$$\left. \frac{\partial c}{\partial z} \right|_l \approx \frac{c_{l+1} - c_{l-1}}{2 \Delta z} \quad , \quad (6)$$

$$\left. \frac{\partial^2 c}{\partial z^2} \right|_l = \left. \frac{\partial \left( \frac{\partial c}{\partial z} \right)}{\partial z} \right|_l \approx \frac{1}{\Delta z} \left[ \frac{c_{l+1} - c_l}{\Delta z} - \frac{c_l - c_{l-1}}{\Delta z} \right] = \frac{c_{l+1} - 2c_l + c_{l-1}}{\Delta z^2} \quad . \quad (7)$$

Im Inneren des Lösungsintervalls läßt sich die partielle Differentialgleichung somit in ein System von  $NZ$  gewöhnlichen Differentialgleichungen überführen. Die Gleichung am Stützpunkt  $l$  lautet dann

$$B \frac{dc_l}{dt} = \alpha c_{l-1} + \beta c_l + \gamma c_{l+1} + S(c_l, z_l, t) \quad . \quad (8)$$

Dabei gilt

$$\alpha = \frac{D}{\Delta z^2} + \frac{v}{2 \Delta z} \quad , \quad (9)$$

$$\beta = \frac{-2D}{\Delta z^2} \quad , \quad (10)$$

$$\gamma = \frac{D}{\Delta z^2} - \frac{v}{2 \Delta z} \quad . \quad (11)$$

Wegen der Kopplung zwischen  $c_l$ ,  $c_{l-1}$  und  $c_{l+1}$  muß das Gleichungssystem simultan gelöst werden. Eine Betrachtung der Lösungsebene  $t$  über  $z$  (Abb. 1) liefert eine

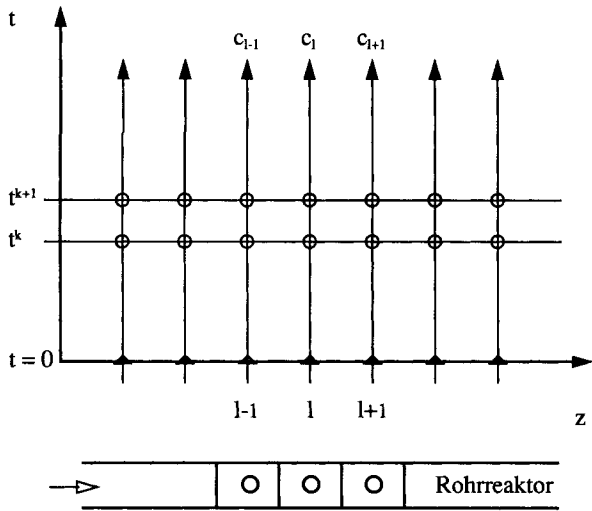


Abb. 1. Zur Ortsdiskretisierung und Zeitintegration von Gl. (1) nach der „method of lines“.

Erklärung für die Bezeichnung „method of lines“: Nach der Ortsdiskretisierung wird jede der entstehenden gewöhnlichen Differentialgl. (8) längs einer Linie  $z = \text{konst.}$  gelöst.

In Vektorschreibweise lautet Gl. (8):

$$B \frac{dy}{dt} = f(y, z, t) \quad (12)$$

mit der Kapazitätsmatrix  $B$ , dem Lösungsvektor  $y$  und dem Funktionenvektor der rechten Seite  $f$ .

$$B = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \ddots & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix} ; \quad y = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \\ \vdots \\ c_{NZ-1} \\ c_{NZ} \end{bmatrix} ; \quad f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_l \\ \vdots \\ f_{NZ-1} \\ f_{NZ} \end{bmatrix} \quad (13)$$

Die Lösung des Systems (12) aus (nichtlinearen) Differentialgleichungen erster Ordnung kann dadurch erfolgen, daß zuerst die Zeitkoordinate  $t$  mit dem Zeitindex  $k$  und der Zeitschrittweite  $\Delta t$  diskretisiert wird:

$$B \frac{y^{k+1} - y^k}{\Delta t} = f^{k+1}(y^{k+1}, z, t + \Delta t) \quad (14)$$

Als nächstes muß die nichtlineare rechte Seite  $f$  linearisiert werden. Dies kann zum Beispiel durch Entwicklung von  $f$  um die alte Lösung  $y^k$  erfolgen. Der entsprechende Ansatz lautet

$$f^{k+1} \approx f^k + \frac{\partial f}{\partial y} (y^{k+1} - y^k) \quad (15)$$

Hierin stellt  $\partial f/\partial y$  die Matrix der partiellen Ableitungen aller rechten Seiten  $f$  nach allen Lösungen  $y$ , die sog. Jacobi-Matrix, zum bekannten Zeitpunkt  $k$  dar.

Die Lösung zum neuen Zeitpunkt  $k+1$  erfordert somit die Lösung des linearen Gleichungssystems

$$\underbrace{\left( \frac{B}{\Delta t} - \frac{\partial f}{\partial y} \right)}_A y^{k+1} = \underbrace{\left( \frac{B}{\Delta t} - \frac{\partial f}{\partial y} \right) y^k + f^k}_b \quad (16)$$

wobei sowohl die Koeffizientenmatrix  $A$  als auch der Vektor der rechten Seite  $b$  aus den Modellkonstanten und den Lösungen des letzten Teilschritts  $y^k$  bestimmt werden können.

Der Lösungsaufwand für das Gleichungssystem (16) hängt ab von der Struktur der Matrix  $A$ , d.h. davon, wieviele Elemente in  $A$  von Null verschieden sind und welches Muster die besetzten Elemente bilden. Es läßt sich leicht zeigen, daß das Muster in der Regel von der Struktur der Jacobi-Matrix  $\partial f/\partial y$  bestimmt wird. Im Einführungsbeispiel ist die Jacobi-Matrix tridiagonal besetzt (Abb. 2a), da gemäß Gl. (8) in der Materialbilanz für den Stützpunkt  $l$  die Konzentrationen  $c_{l-1}$ ,  $c_l$  und  $c_{l+1}$  auftreten. Wenn das Modell des betrachteten Rohrreaktors nicht nur aus einer PDGL für eine Konzentration  $c$  besteht, sondern z.B. Gleichungen für mehrere Konzentrationen sowie Temperatur, Druck und Dichte an jeder Stelle  $l$  benötigt werden, so tritt an die Stelle  $c_l$  im Vektor  $y$  (Glg. (13)) ein Untervektor, bestehend aus den genannten Beschreibungsgrößen an der Stelle  $l$  und an die Stelle der Funktion  $f_l$  im Funktionenvektor  $f$  ein Untervektor der jeweiligen ortsdiskretisierten rechten Seite an der Stelle  $l$ .  $\partial f/\partial y_l$  ist dann die Jacobi-Untermatrix an der Stelle  $l$ , so daß  $\partial f/\partial y$  und damit auch  $A$  eine blocktridiagonale Struktur erhalten (Abb. 2b).

Würde anstelle eines einfach durchströmten Rohrreaktors ein Rohrreaktor mit Rückführung betrachtet, so wäre das erste Element,  $l=1$ , (Zulauf) mit dem letzten Element  $l=NZ$  (Ablauf) verkoppelt. Im Muster von Jacobi-Matrix und  $A$ -Matrix äußert sich das in dem in Abb. 2c zusätzlich vorhandenen Kopplungsblock an der Stelle  $(1, NZ)$ . Auf diese Weise bilden sich die Besonderheiten der Verkopplung und der örtlichen Diskretisierung für jeden Apparat oder Prozeß auf eindeutige Weise in der Matrix  $A$  des linearen Gleichungssystems (16) ab.

Auch die über Gln. (1), (12) und (13) zunächst etwas unmotiviert eingeführte Kapazitätsmatrix  $B$  besitzt für die

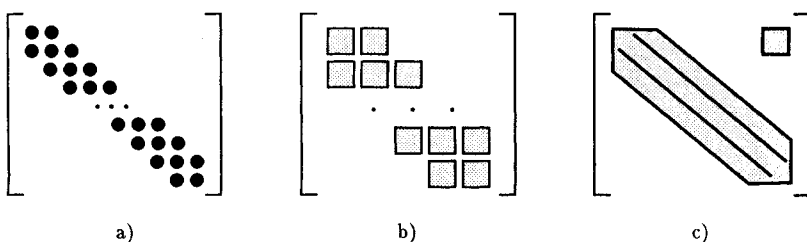


Abb. 2. Übliche Strukturen von Jacobi-Matrix  $J = \partial f/\partial y$  oder Lösungsmatrix  $A$  (Gl. (16)); a) tridiagonale Struktur des Einführungsbeispiels, b) blocktridiagonale Struktur bei Berücksichtigung mehrerer Zustandsgrößen, c) Konvektions-Diffusions-System mit Rückführung.

effiziente und einheitliche numerische Behandlung der Modellgleichungen große Bedeutung. Im Einführungsbeispiel entspricht sie der Einheitsmatrix, häufig ist sie nur diagonal besetzt. Allerdings gibt es Fälle, wo Zeitableitungen mehrerer Beschreibungsgrößen in der gleichen Bilanzgleichung auftreten. So führt z. B. eine Enthalpiebilanz für kompressible Fluide zu einer Summe von Temperatur- ( $\partial T/\partial t$ ) und Gesamtdruckableitung ( $\partial p/\partial t$ ), die in  $B$  ohne zusätzliche Umformung berücksichtigt werden kann.

An Gl. (16) ist erkennbar, daß das Gleichungssystem auch dann lösbar ist, wenn  $B$  nicht regulär ist, also z. B. Nullzeilen enthält. Ein wichtiger Anwendungsfall ist die Gewinnung der stationären Lösung. Hier verschwinden alle Zeitableitungen,  $B$  geht über in die Null-Matrix. Unter diesen Bedingungen beschreibt Gl. (16) ein Iterationsschema für die Gewinnung der stationären Lösung  $y^k$ , nach dem sog. Newton-Raphson-Verfahren (siehe Abschnitt 3), wobei  $k$  jetzt dem Iterationsindex entspricht. Die Lösung des linearen Gleichungssystems (16) muß solange wiederholt werden, bis  $y^{k+1}$  hinreichend gut mit  $y^k$  übereinstimmt.

Außerdem treten bei der Modellformulierung häufig nicht-lineare Kopplungsbeziehungen zwischen den Beschreibungsgrößen auf, die keine Zeitableitung enthalten. Typische Beispiele sind Zustandsgleichungen oder Phasengleichgewichtsbeziehungen, auch als quasistationär betrachtete Bilanzgleichungen gehören dazu. Das heißt, daß Gl. (12) häufig kein System von gewöhnlichen Differentialgleichungen darstellt, sondern Differentialgleichungen und nichtlineare Gleichungen enthält, die über  $f$  und gelegentlich auch  $B$  miteinander verkoppelt sind. Es hat sich eingebürgert, in diesem Fall von Differential-Algebra-(DA)-Systemen zu sprechen.

Ganz allgemein führt also das physikalisch begründete mathematische Modell eines verfahrenstechnischen Prozesses nach einer Diskretisierung der örtlichen Ableitungen auf ein DA-System gemäß Gl. (12). Es kann mit sog. DA-Solvern (vgl. Abschnitt 4) gelöst werden. Das Einführungsbeispiel zeigt dafür eine besonders einfache Vorgehensweise. In allen Fällen resultiert ein lineares Gleichungssystem mit einer Struktur gemäß Gl. (16), das mehrfach (iterativ) gelöst werden muß.

Im folgenden wird auf die einzelnen Schritte

- Diskretisierung in der Ortskoordinate,
  - Diskretisierung in der Zeitkoordinate,
  - Lösung des nichtlinearen Gleichungssystems und
  - Lösung des linearen Gleichungssystems
- in umgekehrter Reihenfolge näher eingegangen.

## 2 Lösung linearer Gleichungssysteme

### 2.1 Direktes Lösungsverfahren (LR-Zerlegung)

Formal erfolgt die Lösung des linearen Gleichungssystems

$$Ay = b \quad (17)$$

durch Bestimmung der Inversen  $A^{-1}$

$$y = A^{-1}b \quad (18)$$

Diese Bestimmung wird allerdings in der Regel aus Gründen des Rechenaufwandes nicht explizit ausgeführt. Vielmehr wird  $A$  durch geeignete Umformung in das Produkt einer Linksdreiecksmatrix  $L$  (Abb. 3a) und einer Rechtsdreiecksmatrix  $R$  (Abb. 3b) zerlegt [1, 3-6].

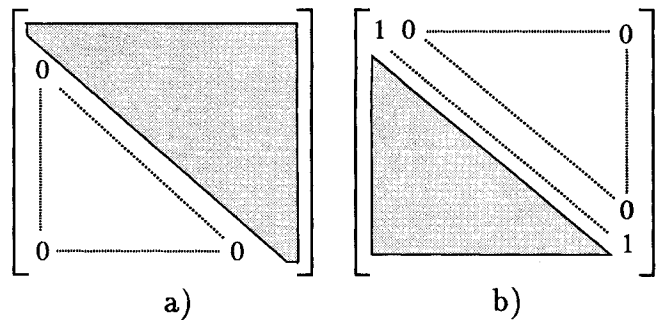


Abb. 3. Aufspaltung von  $A$  gemäß Gl. (19) in a) eine Rechtsdreiecksmatrix  $R$  und b) eine Linksdreiecksmatrix  $L$ .

$$A = LR \quad (19)$$

Sobald  $L$  und  $R$  bekannt sind, läßt sich die Lösung von Gl. (16) oder

$$LRy = b \quad (20)$$

auf zwei einfache Einzelschritte zurückführen: Mit der Substitution  $Ry = x$  löst man zunächst das Gleichungssystem

$$Lx = b \quad (21)$$

durch Vorwärtssubstitution zur Bestimmung des Hilfsvektors  $x$ . Ist dieser bekannt, so folgt der gesuchte Lösungsvektor  $y$  durch Rücksubstitution aus

$$Ry = x \quad (22)$$

Der entscheidende Vorteil dieser Vorgehensweise liegt darin, daß sich die Lösung für beliebige rechte Seiten  $b$  leicht gewinnen läßt, sobald einmal die  $LR$ -Zerlegung von  $A$  durchgeführt wurde. Dieser Vorteil kann bei der numerischen Simulation verfahrenstechnischer Prozesse an verschiedenen Stellen genutzt werden.

Für die  $LR$ -Zerlegung muß auf eine Pivotisierung geachtet werden. Sie ist erforderlich, um bei der Umformung der  $A$ -Matrix Division durch Null oder die Auslöschung signifikanter Stellen durch die Bildung kleiner Differenzen großer Zahlen zu vermeiden. Sie wird meist durch eine vorgeschaltete Skalierung der Beschreibungsgrößen  $y$ , z. B. auf das Intervall  $[-1,1]$ , ergänzt und erleichtert [3, 4, 6].

Für die  $LR$ -Zerlegung existiert eine Vielzahl bewährter Standardroutinen (siehe Abschnitt 2.3), so daß sich der Verfahrenstechniker nur in den seltensten Fällen mit ihrer expliziten Programmierung befassen muß. Ganz allgemein gilt dabei, daß die Struktur der Matrix  $A$  die Lösbarkeit des linearen Gleichungssystems entscheidend beeinflusst. Deshalb ist es sinnvoll, die Modellgleichungen so zu gruppieren, daß die entstehende  $A$ -Matrix in Form von Standardmustern besetzt ist. Dazu zählen insbesondere die in Abb. 2 aufgeführten tridiagonalen oder blocktridiagonalen Strukturen. Sie besitzen den Vorteil, daß ihre Struktur bei der  $LR$ -Zerlegung erhalten bleibt, wenn keine Pivotisierung erforderlich ist.

Gelegentlich kommt es vor, daß die Modellbildung zu keiner der in Abb. 2 gezeigten Standardstrukturen führt, sondern daß die besetzten Elemente sehr unregelmäßig über die Matrix verteilt sind. Das tritt insbesondere bei der Verkopplung unterschiedlicher Apparate oder Prozesse,

d.h. bei der Anlagensimulation auf. Hier, wie ganz allgemein bei örtlich verteilten verfahrenstechnischen Prozessen ist die  $A$ -Matrix nur zu einem kleinen Anteil besetzt; man spricht von schwach besetzten Matrizen („sparse matrices“). Für diese Matrizen existieren spezielle sog. Sparse-Solver, die den Speicherplatz und die notwendigen Rechenoperationen auf die tatsächlich besetzten Elemente der Matrix begrenzen.

## 2.2 Iterative Lösungsverfahren

Die bisher besprochenen Lösungsverfahren für lineare Gleichungssysteme bezeichnet man als direkte Verfahren. Bei weitgehend besetzter Struktur erlauben sie die Lösung von Systemen aus einigen tausend Gleichungen, bei ausgeprägter, diagonaldominanter Bandstruktur auch weit darüber. Da der Rechenaufwand aber mit der Gleichungszahl  $N$  überproportional steigt, besteht eine Obergrenze für  $N$ , ab der eine direkte Lösung weder vom Aufwand noch von den wachsenden Rundungsfehlern her gerechtfertigt ist. Sie hängt neben der Matrixstruktur von der sog. Konditionierung und der verfügbaren Rechnergenauigkeit ab. Diese Schranke kann insbesondere bei der Simulation örtlich zwei- und dreidimensionaler Modellgleichungen leicht erreicht und überschritten werden.

Als Ausweg bieten sich iterative Lösungsverfahren an. Während aber direkte Verfahren bei nicht zu großer Gleichungszahl  $N$  und regulärer Matrix  $A$  stets die Lösung finden, ist die Konvergenz für die klassischen iterativen Verfahren nur bei sog. diagonaldominanten Systemen gesichert. Moderne iterative Verfahren sind in der Lage, durch geeignete Umformungen die Diagonaldominanz zu erzeugen [3, 6].

In einfachster Version eines iterativen Lösungsverfahrens, des sog. Jacobi-Verfahrens, wird die  $n$ -te Gleichung des Gleichungssystems nach der  $n$ -ten Unbekannten aufgelöst und in jedem Iterationsschritt unabhängig von den übrigen Gleichungen gelöst. Für Gl. (17) folgt dann z.B.

$$y_n^{i+1} = \frac{1}{a_{nn}} \left( b_n - \sum_{k=1, k \neq n}^N a_{nk} y_k^i \right) \quad (23)$$

mit den Elementen  $a_{n,k}$  der Matrix  $A$ . Darin kennzeichnet  $i$  den Iterationsindex, d.h. auf der rechten Seite werden jeweils die Unbekannten aus der vorherigen Iteration eingesetzt.

Wenn sie konvergieren, haben iterative Verfahren den Vorteil der Unempfindlichkeit gegen Rundungsfehler und bei schwach besetzten Matrizen den des geringeren Speicherplatzbedarfs.

## 2.3 Software zur Lösung linearer Gleichungssysteme

Für die direkte Lösung von Gleichungssystemen mit vollbesetzter, blocktridiagonaler oder Bandstruktur können die Bibliotheksroutinen von LINPACK [18] oder der IMSL-Bibliothek [19] empfohlen werden. Sie sind wie die übrige hier empfohlene Numerik-Software in Standard-FORTRAN 77 geschrieben und basieren auf  $LR$ -Zerlegung mit Teilpivotisierung.

Für die direkte Lösung von Systemen mit schwach besetzten Matrizen (sparse matrices) hat sich noch kein allgemeiner Standard durchsetzen können. Ein sehr häufig verwendetes Programm ist die Harwell-Routine MA30 (bzw. ihre einfachere zu bedienende Variante MA28) von Duff und

Reid [20, 21], dessen Stärke in der Verwendung der Informationen aus der ersten  $LR$ -Zerlegung für folgende Lösungen von Problemen mit ähnlicher Pivotstruktur liegt. In den letzten Jahren wurden verschiedene neue Algorithmen veröffentlicht [22–24], die aber bei der wiederholten Lösung von Problemen mit identischer Struktur, wie sie bei der Simulation verfahrenstechnischer Prozesse auftritt, meist nicht die Geschwindigkeit von MA30 erreichen [25].

Für die iterative Lösung linearer Gleichungssysteme ist die bei der direkten Lösung erreichte Zuverlässigkeit noch nicht gegeben. In der Praxis muß am Einzelfall untersucht werden, ob ein Verfahren für das konkrete Problem konvergiert oder nicht. Die Entwicklungen der letzten Jahre (z.B. [26]) lassen aber in der nächsten Zeit wesentliche Fortschritte erwarten. Sie werden in speziellen Konvergenzverbesserungsstrategien und in Kombinationen von direkten und iterativen Verfahren gesehen.

Grundsätzlich ist anzumerken, daß Standardverfahren, wie die in diesem Abschnitt beschriebenen, ein Gleichungssystem immer langsamer lösen, als ein für eine konkrete Matrixstruktur optimierter Algorithmus. Allerdings wird sich der Aufwand dafür nur bei sehr zeitkritischen Anwendungen lohnen.

Wie begründet beruht jede numerische Simulation letztlich auf der wiederholten Berechnung linearer Gleichungssysteme. Aus Anwendersicht ist es daher erforderlich, daß die Entwickler von allgemeinen Lösungspaketen für nichtlineare Gleichungen oder DA-Systeme ihre Verfahren von der Lösung der linearen Gleichungssysteme entkoppeln und allgemeine Schnittstellen zur Verfügung zu stellen. Erst dann wird die Verwendung optimaler Lösungsverfahren anstelle universell einsetzbarer mit vertretbarem Aufwand möglich.

## 3 Lösung nichtlinearer Gleichungssysteme

Die iterative Lösung einer nichtlinearen Gleichung in einer Unbekannten

$$0 = f(y) \quad (24)$$

durch das Newton-Verfahren mit dem Iterationsindex  $i$

$$y^{i+1} = y^i - \frac{f(y^i)}{f'(y^i)} \quad (25)$$

ist allgemein bekannt. Die Erweiterung dieser Iterationsvorschrift auf Gleichungssysteme wird als Newton-Raphson-Verfahren bezeichnet und kann formal als Fixpunktiteration geschrieben werden.

$$y^{i+1} = y^i - J(y^i)^{-1} f(y^i) \quad (26)$$

Dabei ist  $J(y)$  die bereits im Zusammenhang mit Gl. (15) eingeführte Jacobi-Matrix

$$J(y) = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \dots & \frac{\partial f_1}{\partial y_n} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \dots & \frac{\partial f_2}{\partial y_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_n}{\partial y_1} & \frac{\partial f_n}{\partial y_2} & \dots & \frac{\partial f_n}{\partial y_n} \end{pmatrix} \quad (27)$$

In der Praxis wird anstelle der Matrixinversion das zu Gl. (26) äquivalente lineare Gleichungssystem

$$J(y^i) \Delta y^i = -f(y^i) \quad (28)$$

mit

$$y^{i+1} = y^i + \Delta y^i \quad (29)$$

iterativ so oft gelöst, bis der Korrekturvektor  $\Delta y^i$  eine vorgegebene Fehlerschranke unterschreitet [1, 3, 5]. Im Einführungsbeispiel wurde in den Gln. (15) und (16) nur der erste Schritt der Newton-Raphson Iteration benutzt, wobei als Startwert der Lösungsvektor zum alten Zeitpunkt  $t^k$  diente.

Der Grund für die häufige Verwendung von direkten oder modifizierten Newton-Raphson-Verfahren liegt im einfachen Algorithmus und in der quadratischen bzw. superlinearen Konvergenz. Allerdings ist die Konvergenz nur bei hinreichend guten Startwerten gesichert. Die Lösung großer nichtlinearer Gleichungssysteme kann deshalb auf erhebliche Schwierigkeiten stoßen, zumal der Algorithmus bei Existenz mehrfacher Lösungen nicht notwendigerweise gegen die gewünschte oder eine physikalisch sinnvolle Lösung konvergiert. Im Zusammenhang mit der Lösung von DA-Systemen treten diese Konvergenzprobleme allerdings nur selten in Erscheinung, da der vorherige Zeitschritt gute Startwerte für die Iteration liefert, wobei die Änderung  $\Delta y$  durch Reduzierung der Zeitschrittweite beliebig verkleinert werden kann.

Ausnahmen von dieser Regel sind im wesentlichen in zwei Fällen zu erwarten: bei Programmstart mit inkonsistenten Startwerten (siehe Abschnitt 4) und bei Funktionen, die in der Nähe der Lösung ein lokales oder globales Minimum besitzen. Sehr leicht ist die Problematik im Fall einer Unbekannten zu erkennen. Die Gleichung  $0 = y^2$  hat eine (doppelte) Nullstelle bei  $y = 0$ . Gleichzeitig liegt dort das globale Minimum der Funktion, es gilt somit  $f'(y = 0) = 0$ . Die Iterationsvorschrift (25) führt dann auf eine Division durch Null.

Desweiteren können Schwierigkeiten auftreten, wenn die Funktion  $f$  nicht überall definiert ist, da das Newton-Raphson-Verfahren mitunter nicht monoton, sondern oszillierend um die Lösung konvergiert. Ist die Funktion in der Nähe der Nullstelle nicht definiert, besteht die Gefahr, daß die Funktion in einem nichtdefinierten Bereich ausgewertet werden muß. Ein einfaches Beispiel liefert die Reaktionsgeschwindigkeit  $r = c^{0.5}$ . Der naheliegende Gedanke, der Reaktionsgeschwindigkeit für negative Konzentrationen den Wert Null zuzuordnen, kann die Konvergenz des Newton-Raphson-Verfahren verhindern, weil dann die Ableitung der Funktion  $f$  in der Nähe der Lösung nicht mehr stetig ist. Ein sinnvoller Ausweg ist in diesen Fällen die stetige Fortsetzung der Funktion in den nicht definierten Bereich, z.B. durch  $r(c < 0) = -|c|^{0.5}$ .

Bei der Bestimmung der Jacobi-Matrix ist zunächst die Frage nach analytischer oder näherungsweise numerischer Berechnung zu stellen. In aller Regel liefert die Auswertung der analytischen Ableitungen das genauere und meist auch das schnellere Ergebnis. Allerdings ist das analytische Ableiten von komplexen Funktionen mit erheblichem Aufwand und der Gefahr von Fehlern verbunden. Beispiele dafür sind implizite Beziehungen zur Stoffdatenberechnung für Mehrkomponentensysteme oder komplexe Reaktionsgeschwindigkeitsansätze. Aus diesem Grund besitzen viele Bibliotheksprogramme die Option, die Jacobi-Matrix gemäß

$$\frac{\partial f_i}{\partial y_j} \approx \frac{f_i(y_j + \Delta y_j) - f_i(y_j)}{\Delta y_j} \quad (30)$$

numerisch zu berechnen. Dabei muß durch geeignete Skalierung und Wahl der Schrittweite  $\Delta y_j$  einer Auslöschung von signifikanten Stellen entgegengewirkt werden. Durch Ausnutzung der in der Regel schwach besetzten Struktur der Jacobi-Matrix läßt sich die Anzahl der Funktionsaufrufe zur Berechnung von  $f(y + \Delta y)$  stark reduzieren, indem vor jedem Funktionsaufruf mehrere Komponenten von  $f$  verändert werden. So reichen in unserem Einführungsbeispiel zur vollständigen Berechnung der triagonalen Jacobi-Matrix unabhängig von der Stützstellenzahl vier Funktionsauswertungen aus.

Nachdem in den letzten Jahren leistungsfähige Programme zur symbolischen Rechnung verfügbar geworden sind (z.B. MAPLE [35] oder MACSYMA [36]), können analytische Ableitungen aber heute einfach und sicher bestimmt werden. Das Ergebnis in Form von FORTRAN-Code läßt sich direkt in den Gleichungslöser übernehmen.

Bei der Iteration nach (26) treten vor allem zwei Kostenfaktoren auf:

- Die Lösung des linearen Gleichungssystems und
- Die Berechnung von Jacobi-Matrix  $J$ .

Auf effiziente Verfahren zur Lösung linearer Gleichungssysteme sind wir im vorigen Abschnitt eingegangen. Die einfachste Strategie zur Verminderung des Aufwands bei der Berechnung der Jacobi-Matrix besteht darin, die Jacobi-Matrix während mehrerer Iterationen konstant zu lassen. Dadurch werden zwar die Konvergenzordnung und der Konvergenzradius verringert, dafür fällt aber neben dem Aufwand für die Bestimmung der Jacobi-Matrix auch der für die LR-Zerlegung weg. Es hängt jeweils vom Einzelfall ab, wieweit diese Möglichkeit sinnvoll genutzt werden kann.

Ein anderer Weg besteht darin, nicht die gesamte Jacobi-Matrix neu zu berechnen, sondern die anfangs bestimmte in jedem Iterationsschritt zu verbessern. Entsprechende Verfahren beruhen meist auf der von Broyden [28] entwickelten Methode. Ihr Nachteil ist die Erzeugung zusätzlicher Nicht-Null-Elemente in der Matrixstruktur. Verschiedene Autoren haben sich mit dem Problem der Broydenupdates bei weitgehender Erhaltung der Matrixstruktur beschäftigt [29, 30].

Eine zweckmäßige Möglichkeit zur Beeinflussung von Konvergenzgeschwindigkeit und Konvergenzradius bei Newton-Verfahrens ist die sog. Relaxation gemäß

$$y^{i+1} = y^i + \omega \Delta y^i \quad (31)$$

Dabei wird ein Schritt mit  $\omega > 1$  als Überrelaxation, mit  $\omega < 1$  als Unterrelaxation bezeichnet. Die Überrelaxation wird zur Erhöhung der Konvergenzgeschwindigkeit eingesetzt. Praktisch bedeutsam ist vor allem die Unterrelaxation, die den Konvergenzradius erheblich vergrößern kann [3, 34].

### 3.1 Standardsoftware zur Lösung nichtlinearer Gleichungssysteme

In allen großen mathematischen Standardbibliotheken (wie IMSL oder Harwell) sind Newton- und Quasi-Newton-Verfahren zur Berechnung der Nullstellen nichtlinearer Gleichungssysteme enthalten. Besonders bewährt hat sich bei uns bei kritischen Problemen das Programm NLEQ1 von Deufthard [31], unter anderem bei der Berechnung von simultanen Reaktionsgleichgewichten sowie zur

Initialisierung von DA-Systemen (s. Abschnitt 4.2). Es besitzt eine sehr effektive Steuerung des Relaxationsparameters  $\omega$ . Neuere Übersichten und Vergleiche verschiedener Verfahren zur Lösung nichtlinearer Gleichungssysteme aus der Sicht des Chemieingenieurs stammen von *Shacham* [32] und *Sun* [33].

#### 4 Integration von Differentialgleichungen

Das bei der Modellbildung gewonnene System besteht im allgemeinen aus partiellen Differentialgleichungen, gewöhnlichen Differentialgleichungen erster und höherer Ordnung und algebraischen Gleichungen. Mit Hilfe der „method of lines“ können die partiellen Differentialgleichungen auf ein System gewöhnlicher Differentialgleichungen zurückgeführt werden. Die Umwandlung gewöhnlicher Differentialgleichungen höherer Ordnung in ein System von Differentialgleichungen erster Ordnung erfolgt durch einfache Substitution.

Zunächst sei die Integration einer gewöhnlichen Differentialgleichung erster Ordnung mit allgemeiner nichtlinearer rechter Seite  $f(y, t)$  betrachtet.

$$y' = \frac{dy}{dt} = f(y, t) \quad (32)$$

Die einfachste Möglichkeit besteht in der Approximation des Differentialquotienten  $dy/dt$  durch einen Differenzenquotienten, z.B.

$$\frac{dy}{dt} = \frac{y^{k+1} - y^k}{\Delta t} = f(y, t) \quad (33)$$

Dabei ist die Frage nach einer geeigneten Annäherung der Funktion  $f(y, t)$  im Intervall  $[t^k, t^k + \Delta t]$  zu stellen. Einen flexiblen Ansatz dafür liefert das allgemeine Euler-Verfahren (Abb. 4).

$$\frac{y^{k+1} - y^k}{\Delta t} = (1 - a)f(y^k, t^k) + a f(y^{k+1}, t^k + \Delta t) \quad (34)$$

Mit  $a = 0$ , d.h.  $f(y, t) \approx f(y^k, t^k)$  erhält man das sog. explizite Euler-Verfahren, das nach  $y^{k+1}$  aufgelöst werden kann.

$$\frac{y^{k+1} - y^k}{\Delta t} = f(y^k, t^k) \quad (35)$$

Im Gegensatz dazu führt das sog. implizite Euler-Verfahren mit  $a = 1$ , d.h.  $f(y, t) \approx f(y^{k+1}, t^k + \Delta t)$ , im allgemeinen

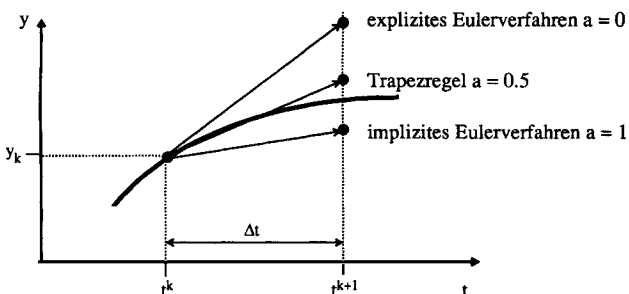


Abb. 4. Integration von Gl. (32) nach dem allgemeinen Euler-Verfahren (Gl. (34)).

auf eine nichtlineare Beziehung, die iterativ mit den im vorigen Abschnitt beschriebenen Methoden gelöst werden muß.

$$\frac{y^{k+1} - y^k}{\Delta t} = f(y^{k+1}, t^k + \Delta t) \quad (36)$$

Beide Verfahren erreichen eine Genauigkeit erster Ordnung. Für  $a = 1/2$  folgt aus Gl. (34) die sog. Trapezregel, die, da  $f^{k+1}$  benötigt wird, ebenfalls implizit ist, aber eine Genauigkeit zweiter Ordnung erreicht.

Dabei bedeutet Genauigkeit  $n$ -ter Ordnung, daß sich der Diskretisierungsfehler bei der Lösung der Differentialgleichung proportional zu  $\Delta t^n$  verhält. Mit einem Verfahren höherer Ordnung läßt sich der Diskretisierungsfehler also durch Verkleinerung von  $\Delta t$  stärker reduzieren.

Mit einer Zentraldifferenzenapproximation der Zeitableitung ließe sich das folgende explizite Verfahren erzeugen, das ebenfalls von zweiter Ordnung ist

$$\frac{y^{k+1} - y^{k-1}}{2 \Delta t} = f(y^k, t^k) \quad (37)$$

Allerdings zeigt eine Analyse, daß durch diese Diskretisierung eine instabile parasitäre Lösung eingeschleppt wird. Sie führt unabhängig von der Wahl von  $\Delta t$  nach mehreren Schritten zu aufklingenden Oszillationen um die tatsächliche Lösung, so daß sich Gl. (37) als unbrauchbar erweist [7].

Notwendige Voraussetzung für einen Lösungsalgorithmus ist also zunächst eine stabile Approximation des zeitlichen Differentialquotienten. Diese Voraussetzung ist beim allgemeinen Euler-Verfahren erfüllt. Trotzdem kann es auch hier zu unerwünschten Oszillationen kommen, wenn  $a < 1$  gilt und die Zeitschrittweite bestimmte Grenzen überschreitet. Das wird in Abb. 5 am Beispiel der stabilen linearen Differentialgleichung

$$\tau \frac{dy}{dt} = -y \quad (38)$$

mit der Lösung

$$y(t) = y^0 \exp(-t/\tau) \quad (39)$$

gezeigt. Anhand von Gln. (35) und (36) ist der numerische Lösungsverlauf für große Zeitschrittweiten leicht zu verfolgen.

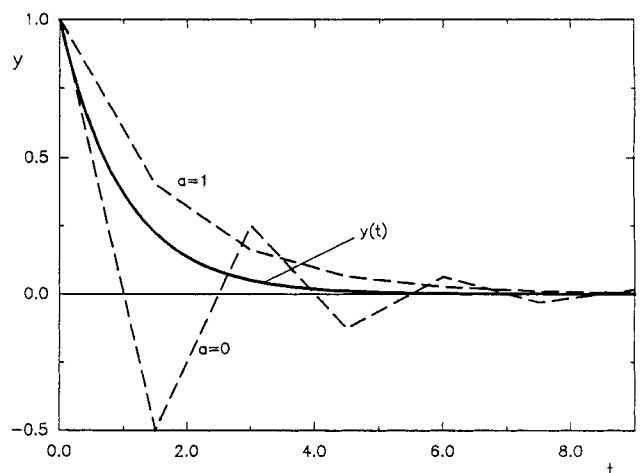


Abb. 5. Numerische Lösung von Gl. (38) mit  $\Delta t = 1.5 \tau$  mit dem impliziten ( $a = 1$ ) und dem expliziten ( $a = 0$ ) Euler-Verfahren.

gen. Es zeigt sich, daß im Fall des expliziten Euler-Verfahrens die Lösung zu oszillieren beginnt, sobald im Beispiel  $\Delta t > \tau$  ist, und exponentiell aufklingt, sobald  $\Delta t > 2 \tau$  ist. Das implizite Euler-Verfahren nähert sich dagegen auch für beliebig große Zeitschritte  $\Delta t$  der exakten Lösung monoton an.

Dieses Ergebnis läßt sich in bemerkenswerter Weise verallgemeinern: Alle rein expliziten Integrationsverfahren werden bei der Lösung stabiler Differentialgleichungen oberhalb einer Grenzschrittweite  $\Delta t_{\text{krit}}$  instabil, wobei  $\Delta t_{\text{krit}}$  in der Größenordnung der kleinsten Zeitkonstanten  $\tau$  bzw. des betragsmäßig kleinsten reziproken Eigenwertes der linearisierten Differentialgleichung liegt. Hingegen erlauben implizite Lösungsverfahren auch Zeitschritte, die wesentlich größer als  $\tau$  sind. Dieses Ergebnis besitzt entscheidende Bedeutung für die Lösung von Differentialgleichungssystemen (s. Abschnitt 4.1). Hingegen gibt es keinen Grund, bei der Lösung einer einzelnen Differentialgleichung erster Ordnung implizite Verfahren den expliziten vorzuziehen. Um eine genaue Lösung zu erzielen, muß die Schrittweite ohnehin deutlich kleiner als die Zeitkonstante  $\tau$  gewählt werden und der Rechenaufwand expliziter Verfahren ist naturgemäß wesentlich geringer. Deswegen wurden in der Vergangenheit auch hauptsächlich explizite Lösungsverfahren für gewöhnliche Differentialgleichungen eingesetzt. Das populärste Beispiel ist das klassische Runge-Kutta-Verfahren 4. Ordnung, andere häufig verwendete Methoden sind das Adams-Moulton- und das Adams-Bashford-Verfahren [8].

Für eine effiziente Nutzung von Integrationsverfahren ist eine automatisierte Steuerung der Zeitschrittweite unerläßlich. Neben der Sicherung der Stabilität (bei expliziten Verfahren) und der Konvergenz (bei impliziten Verfahren) sorgt sie für eine Kontrolle des Fehlers und eine Minimierung der zur Lösung einer Aufgabe benötigten Rechenzeit. In aller Regel basiert die Schrittweitensteuerung auf einer Fehlerabschätzung, z. B. aus einem Vergleich von Ergebnissen mit unterschiedlicher Schrittweite  $\Delta t$ .

#### 4.1 Integration eines Systems von gewöhnlichen Differentialgleichungen

Bei der Integration eines Systems von gewöhnlichen Differentialgleichungen

$$By' = f(y, t) \quad (40)$$

(mit regulärer Matrix  $B$ ) kann die Verwendung eines der oben beschriebenen expliziten Verfahren zu erheblichen Problemen führen. Ein kurzes Beispiel soll dies verdeutlichen. Zu berechnen sei der Konzentrationsverlauf im isothermen Batchreaktor mit konstantem Volumen für eine einfache Folgereaktion:



Der zeitliche Verlauf der Konzentrationen wird durch die nachstehenden Differentialgl. erster Ordnung beschrieben.

$$\frac{dc_A}{dt} = -k_1 c_A \quad (42)$$

$$\frac{dc_B}{dt} = k_1 c_A - k_2 c_B \quad (43)$$

$$\frac{dc_C}{dt} = k_2 c_B \quad (44)$$

Die analytische Lösung dieses Systems lautet

$$c_A = c_{AO} \exp(-k_1 t) \quad (45)$$

$$c_B = \frac{k_1 c_{AO}}{k_2 - k_1} (\exp(-k_1 t) - \exp(-k_2 t)) \quad (46)$$

$$c_C = \frac{c_{AO}}{k_2 - k_1} (-k_2 \exp(-k_1 t) + k_1 \exp(-k_2 t)) + c_{AO} \quad (47)$$

Dabei ist  $c_{AO}$  die Konzentration des Edukts A zum Zeitpunkt  $t = 0$ . Läuft Reaktion 2 sehr viel schneller ab als Reaktion 1, dann klingt der Anteil  $\exp(-k_2 t)$  schnell ab; das dynamische Verhalten des Systems wird dann nur von  $\exp(-k_1 t)$  bestimmt. Abb. 6 zeigt die Konzentrationsverläufe für die Zahlenwerte  $k_1 = 1$  und  $k_2 = 10^6$ . Man erkennt, daß das Übergangsverhalten nach einem sehr

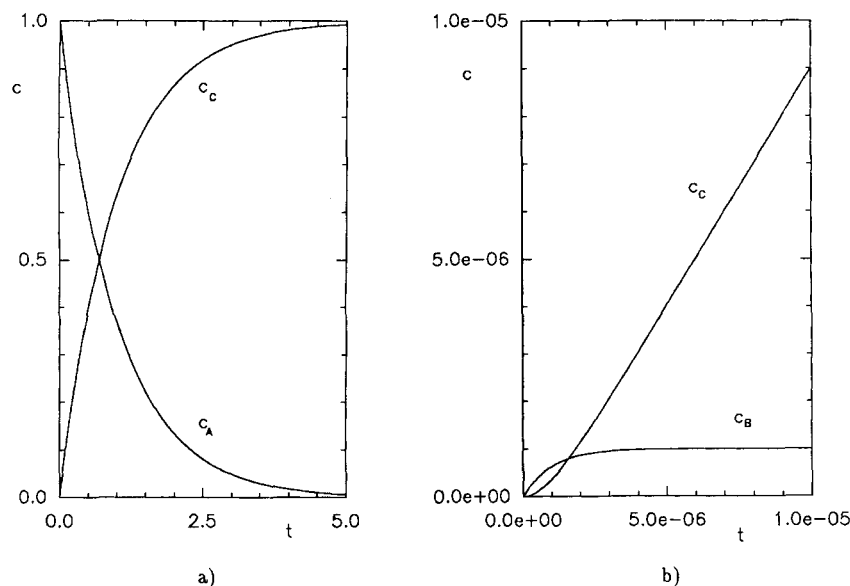


Abb. 6. a) Konzentrationsverläufe der Folgereaktion (Gl. (41)) für  $k_1 = 1$ ,  $k_2 = 10^6$ ; b) Kurzzeitdynamik der Folgereaktion.



kurzen Einschwingvorgang nahezu ausschließlich von der langsamen Dynamik mit  $\tau_1 = 1/k_1 = 1$  bestimmt wird. Trotzdem muß bei der Integration mit einem expliziten Verfahren die Zeitschrittweite aus den genannten Gründen in der Größenordnung der kleinsten Zeitkonstanten, d.h. von  $\tau_2 = 1/k_2 = 10^{-6}$  (!) gewählt werden, damit die Lösung stabil bleibt. Ein solches System wird als „steif“ bezeichnet. Als Maß für die Steifheit kann allgemein das Verhältnis von kleinstem und größtem Realteil der Eigenwerte des Differentialgleichungssystems bezeichnet werden [5]. Die Integration eines steifen Systems der Form (40) mit einem expliziten Verfahren ist nicht mehr effektiv: die Zahl der nur der Stabilität, nicht aber der Genauigkeit wegen erforderlichen Integrations Schritte wird so groß, daß eine Integration nur noch mit impliziten Verfahren sinnvoll möglich ist.

Bei nichtlinearen Differentialgl. folgen die Eigenwerte aus den um die momentane Lösung linearisierten Gleichungen. Daher kann sich die Steifheit im Verlauf der Lösung stark ändern. Der kleinste Eigenwert eines durch Ortsdiskretisierung einer PDGL entstehenden Gleichungssystems hängt außerdem von der gewählten Ortsschrittweite ab. Aus diesen und den in Abschnitt 4.2 genannten Gründen empfiehlt es sich, für die allgemeine Lösung von verfahrenstechnischen Problemen nach der „method of lines“ von vornherein auf explizite Integrationsverfahren zu verzichten.

Das im Einführungsbeispiel verwendete sog. semi-implizite Euler-Verfahren [37]

$$B \frac{y^{k+1} - y^k}{\Delta t} = f(y^k) + \left. \frac{\partial f}{\partial y} \right|_{y^k} (y^{k+1} - y^k) \quad , \quad (48)$$

das aus dem impliziten Euler-Verfahren (36) durch Linearisieren des Terms  $f^{k+1}$  um den alten Zeitpunkt  $t^k$  gewonnen wird, führt zu der formal expliziten Integrationsvorschrift

$$\left( B - \Delta t \left. \frac{\partial f}{\partial y} \right|_{y^k} \right) y^{k+1} = \left( B - \Delta t \left. \frac{\partial f}{\partial y} \right|_{y^k} \right) y^k + \Delta t f(y^k) \quad , \quad (49)$$

ohne dabei die Stabilitätseigenschaften des impliziten Euler-Verfahrens zu verlieren [17]. Damit ist es auch für steife Systeme geeignet. Es bildet die Basis für LIMEX [47], ein effizientes Standardverfahren zur Lösung von DA-Systemen (siehe Abschnitt 4.4).

## 4.2 Integration von Differential-Algebra-Systemen

Im allgemeinen besteht das Modell eines verfahrenstechnischen Apparates aus Differentialgleichungen, die sein dynamisches Verhalten beschreiben, und aus algebraischen Kopplungen (DA-System). Es kann ebenfalls in der Form (40) geschrieben werden, die Matrix  $B$  ist dann singulär (sie enthält leere Zeilen und/oder Spalten). Typische Beispiele für lineare und nichtlineare Kopplungsbeziehungen sind die Randbedingungen partieller Differentialgleichungen, Phasengleichgewichte, Zustandsgleichungen und als quasistationär angenommene partielle Differentialgleichungen. DA-Systeme können, wenn ihr Index  $k_d \leq 1$  ist (s. folgender Abschnitt), wie steife Systeme mit impliziten Integrationsverfahren gelöst werden [38]. Anschaulich kann man algebraische Beziehungen als Differentialgleichungen mit unendlich kurzer Einschwingzeit auffassen, DA-Systeme sind dann unendlich steife Differentialgleichungssysteme.

Für die numerische Lösung von DA-Systemen müssen Anfangsbedingungen spezifiziert werden. Dabei kann die Anzahl der frei wählbaren Anfangsbedingungen (die „Freiheitsgrade des DA-Systems“) kleiner sein als die Anzahl der Variablen, deren zeitliche Ableitung das System enthält. Darauf wird im folgenden Abschnitt näher eingegangen. In jedem Fall müssen vor dem Start der Integration die übrigen (nicht frei wählbaren) Variablen in einem Initialisierungsschritt so bestimmt werden, daß sie das DA-System zum Startzeitpunkt erfüllen. Die meisten Standardprogramme zur Integration von DA-Systemen weisen nämlich schlecht initialisierte Systeme zurück. Der Initialisierungsschritt erfordert für  $k_d \leq 1$  die Lösung der algebraischen Gleichungen des DA-Systems. Falls keine guten Schätzwerte zur Verfügung stehen, kann die Initialisierung zu Konvergenzproblemen führen. Dann ist es erforderlich, z.B. über Näherungsbetrachtungen geeignete Schätzwerte zu bestimmen.

Ein anderer Ausweg besteht darin, die Integration des DA-Systems mit einem impliziten Euler-Schritt (Gl. (36)) und vollständiger Newton-Raphson-Iteration zu beginnen. Wenn die Iteration konvergiert, liefert sie einen konsistenten Startwertsatz. Bei Index-Problemen ( $k_d \geq 2$ ) ändern sich allerdings in diesem Startschritt häufig auch die gewählten Anfangsbedingungen in physikalisch nicht sinnvoller Weise. Dann ist die im folgenden beschriebene Umformung des DA-Systems erforderlich.

## 4.3 Index-Problematik

In den meisten Fällen können DA-Systeme mit impliziten oder semiimpliziten Integrationsverfahren problemlos gelöst werden. In einigen Fällen macht sich jedoch eine Struktureigenschaft der DA-Systeme, der sog. Index, negativ bemerkbar. Zur Definition des Index existieren unterschiedliche Aussagen, die am weitesten verbreitete ist die des differentiellen Index [40, 41]. Demnach entspricht der differentielle Index  $k_d$  eines DA-Systems der Anzahl der Ableitungen die notwendig sind, um das DA-System in ein System gewöhnlicher Differentialgleichungen zu überführen. DA-Systeme mit einem Index von  $k_d \leq 1$  können mit den oben beschriebenen Methoden für steife Differentialgleichungssysteme integriert werden. Die Integration von Systemen mit  $k_d \geq 2$  führt hingegen auf zahlreiche Probleme, so daß Standardmethoden häufig versagen. Zur Veranschaulichung betrachten wir das folgende einfache Beispiel:

$$y_1' = y_1 + 2 y_2 + 2 z \quad , \quad (50)$$

$$y_2' = y_1 - y_2 - z \quad , \quad (51)$$

$$0 = y_1 + y_2 \quad . \quad (52)$$

Es zeichnet sich durch die Besonderheit aus, daß  $y_1$  und  $y_2$  in Gl. (52) algebraisch verkoppelt sind. Offensichtlich kann daher nur eine der beiden als Anfangsbedingung frei vorgegeben werden, während der Startwert der anderen zusammen mit dem der Variablen  $z$  in einem Initialisierungsschritt zu bestimmen ist. Allerdings ist diese Initialisierung nicht sofort möglich, da keine explizite Beziehung zur Berechnung von  $z$  vorliegt. Zu ihrer Bestimmung kann zunächst Gl. (52) abgeleitet werden:

$$y_1' + y_2' = 0 \quad . \quad (53)$$

Nach Einsetzen von (50) und (51) in (53) erhält man die im ursprünglichen System versteckte algebraische Beziehung

$$0 = 2 y_1 + y_2 + z \quad , \quad (54)$$

aus der sich der Initialisierungswert von  $z$  berechnen läßt. Wird Gl. (54) abgeleitet, folgt nach Einsetzen von (50) und (51) die nachstehende Differentialgl. für  $z$

$$z' = -3(y_1 + y_2 + z) \quad . \quad (55)$$

Da zur Umwandlung des ursprünglichen DA-Systems (50) bis (52) in das System gewöhnlicher Differentialgleichungen (50), (51), (55) zwei Ableitungen erforderlich waren, beträgt der differentielle Index des ursprünglichen Systems 2.

Zur allgemeinen Behandlung von Indexproblemen mit ( $k_d \geq 2$ ) schlagen *Bachmann* u.a. [45, 46] vor, aus dem ursprünglichen DA-System durch Differenzieren und Einsetzen die versteckten algebraischen Kopplungen zu entwickeln und die algebraischen Gleichungen zusammen mit der notwendigen Zahl von Differentialgl. zu lösen. Im obigen Beispiel besteht das Ersatzsystem aus den Gln. (52) und (54), sowie einer der drei Differentialgl. (50), (51), (55) und besitzt den differentiellen Index  $k_d = 1$ .

Dies zeigt, daß von der Struktur des DA-Systems her alle drei Variablen  $y_1$ ,  $y_2$  und  $z$  gleichberechtigt sind und es freigestellt ist, einer von ihnen eine Anfangsbedingung (einen Freiheitsgrad) zuzuordnen. In der Praxis wird es sich nach der physikalischen Aufgabenstellung richten, für welche Variable man die Anfangsbedingung vorgibt.

Die Umformung des ursprünglichen DA-Systems durch Differenzieren und Einsetzen kann bei komplexen Systemen sehr aufwendig werden. Dann bietet es sich an, die zu eliminierende Zeitableitung einer Variablen durch eine neue (algebraische) Unbekannte zu ersetzen, wobei die Ableitungen der algebraischen Kopplungsbedingungen die zusätzlichen Bestimmungsgleichungen liefern. Im obigen Beispiel führt die Substitution von  $y_2' = z_2$  mit Gl. (53) auf das äquivalente DA-System mit Index  $k_d = 1$ .

$$y_1' = y_1 + 2 y_2 + 2 z \quad , \quad (56)$$

$$0 = y_1 - y_2 - z - z_2 \quad , \quad (57)$$

$$0 = y_1 + y_2 \quad , \quad (58)$$

$$y_1' = -z_2 \quad . \quad (59)$$

Durch diese Vorgehensweise wird zwar das ursprüngliche System durch neue Unbekannte vergrößert, der Umformungsaufwand aber erheblich reduziert.

Bei dynamischen Modellen verfahrenstechnischer Prozesse kann die Indexproblematik immer dann auftreten, wenn Zustandsgrößen über Zustandsgleichungen, Phasengleichgewichtsbeziehungen oder spezielle Transportansätze miteinander verkoppelt sind. Dabei hat sich herausgestellt, daß eine Indexproblematik bereits bei der Formulierung der zugrunde liegenden partiellen Differentialgleichungen vorliegen kann [42]. Wie in diesem Fall, also vor der Ortsdiskretisierung nach der „method of lines“ zu verfahren ist, ist Gegenstand laufender Untersuchungen.

#### 4.4 Standardsoftware für DA-Systeme

Mit den in den großen Programmbibliotheken zur numerischen Mathematik (z.B. IMSL, NAG, Harwell) enthalte-

nen Integrationsroutinen können einfache Probleme sicher und schnell gelöst werden. Effektiver ist aber im allgemeinen der Einsatz von Programmen, die neuere Ergebnisse der mathematischen Forschung implementieren – gerade auf dem Gebiet der numerischen Integration von DA-Systemen wurden in den achtziger Jahren entscheidende Fortschritte gemacht.

Als sehr leistungsfähig erwiesen sich dabei die folgenden Methoden:

- Backward-Differential-Formula-(BDF)-Verfahren,
- Extrapolations-Verfahren und
- implizite Runge-Kutta-Verfahren.

Die weiteste Verbreitung haben die auf BDF-Verfahren beruhenden Programme DASSL [43] und LSODE/LSODI [44] gefunden. Die von Deuffhard entwickelten Extrapolationscodes (z.B. EULSIM und LIMEX [47]) haben sich in unseren Anwendungen als außerordentlich leistungsfähig und in Problemen von praktischer Relevanz DASSL durchaus gleichwertig gezeigt. Stark zugenommen hat in den letzten Jahren das Interesse an impliziten Runge-Kutta-Verfahren [48–50]. Die dabei entwickelten Codes (z.B. RADAU5 [50]) haben noch nicht den Stand der oben erwähnten Programme erreicht, in den nächsten Jahren kann hier aber mit wesentlichen Fortschritten gerechnet werden.

Alle erwähnten Programme enthalten eine automatische Fehler- und Schrittweitensteuerung, LIMEX ist darüberhinaus in der Lage, Probleme mit einem Index  $k_{d \geq 2}$  zu erkennen und automatisch zurückzuweisen. Im allgemeinen werden diese Routinen ein Problem vergleichbar schnell lösen, es muß am Einzelfall untersucht werden, welches Verfahren am besten geeignet ist. Falls in einem System häufig sog. Schaltvorgänge auftreten, die sich in Unstetigkeiten der rechten Seite  $f(y, t)$  des mathematischen Modells äußern, sind Einschrittverfahren (wie Extrapolations- oder Runge-Kutta-Verfahren) vorzuziehen, weil ein Neustart bei Mehrschrittverfahren (wie BDF) zunächst nur mit niedriger Ordnung erfolgt.

Sog. simultane Verfahren der dynamischen Anlagensimulation wie DIVA [5] oder SPEEDUP [52] basieren auf einem leistungsfähigen DA-Solver, wobei aus den ortsdiskretisierten DA-Systemen der einzelnen Anlagenteile ein der Gl. (40) entsprechendes Gesamtsystem der ganzen Anlage erstellt und mit den vorstehend beschriebenen Methoden gelöst wird.

## 5 Integration partieller Differentialgleichungen

Wie im Einführungsbeispiel gezeigt, lassen sich partielle Differentialgleichungssysteme durch Ortsdiskretisierung in DA-Systeme umformen. Während für die Lösung von DA-Systemen das beschriebene umfangreiche und erprobte Instrumentarium existiert, das eine Lösung mit vorzuziehender Genauigkeit gestattet, ist die Methodik zur Ortsdiskretisierung nicht so weit entwickelt. Insbesondere muß der Verfahrens-Ingenieur die Ortsdiskretisierung auch bei kommerzieller Simulationssoftware in der Regel selbst durchführen, will er nicht mit vorgefertigten Apparatemodulen mit eingeschränktem Anwendungsbereich vorlieb nehmen. In aller Regel wird dabei der durch die Ortsdiskretisierung entstehende Fehler nicht näher betrachtet und begrenzt.

Zur Diskretisierung der Ortsableitungen existiert eine Vielzahl von Methoden. Am bekanntesten für verfahrens-

technische Anwendungen sind Finite-Differenzen-Verfahren und Finite-Elemente-Methoden. Im örtlich eindimensionalen Fall führen beide auf identische oder ähnliche Gleichungssysteme. Wegen der einfachen Ableitung wird den Differenzverfahren im eindimensionalen, mitunter auch im zweidimensionalen Fall der Vorzug gegeben. Die im Einführungsbeispiel in den Gln. (5) bis (7) durchgeführte Differenzenapproximation sei als Beispiel genannt.

Auf eine ausführliche Darstellung der bei der Ortsdiskretisierung auftretenden Probleme und ihre zweckmäßige Lösung muß in diesem Beitrag verzichtet werden; sie ist im Rahmen einer getrennten Veröffentlichung geplant. Hier sei nur auf einige einführende und zusammenfassende Monographien verwiesen [9, 10]. Einführende Kapitel in die Behandlung partieller Differentialgleichungen finden sich auch in einigen Standardbüchern zur numerischen Mathematik [5, 6]. Für die Lösung verfahrenstechnischer Probleme gebräuchliche Verfahren werden in [11, 12] angesprochen. Zur Einführung in sog. Finite-Volumen-Verfahren sei auf [13] und in Finite-Elemente-Verfahren auf [14–16] verwiesen.

## 6 Informations- und Codequellen

Der vorstehende Überblick über numerische Verfahren für die Simulation des dynamischen und stationären Verhaltens verfahrenstechnischer Prozesse sollte eine Vorgehensweise referieren, die sich in den letzten Jahren etabliert und bewährt hat. Genauere Informationen sind der jeweils genannten Fachliteratur zu entnehmen. Es sollte außerdem deutlich gemacht werden, daß für die Grundoperationen der Lösung linearer Gleichungssysteme, der Lösung nichtlinearer Gleichungssysteme und der Lösung von DAE-Systemen bewährte Standard-Code existieren, so daß eine eigene Programmierung dieser Schritte auf wenige Sonderfälle beschränkt bleiben kann.

Die Standard-Codes sind in der Regel als FORTRAN 77-Unterprogramme zum einen in den großen, kommerziell vertriebenen, lizenzpflichtigen numerischen Bibliotheken wie IMSL, Harwell oder NAG enthalten. Zum anderen ist in den letzten Jahren eine Reihe interessanter Codes als „public domain“-Software frei verfügbar geworden. Das heißt, daß diese Unterprogramme jederzeit frei kopiert und in eigenen Systemen verwendet werden dürfen, wie zum Beispiel die LINPACK-Sammlung für die Lösung linearer Gleichungssysteme zusammen mit den gleichzeitig entwickelten BLAS-Routinen (BLAS = Basic Linear Algebra Subroutines) für die grundlegenden Vektor- und Matrix-Operationen. An den meisten Hochschulrechenzentren in Deutschland sind diese Lösungen vorhanden und können kostenfrei kopiert werden. Weitere allgemeine und spezielle Codes werden an verschiedenen Orten zentral gesammelt. Der bekannteste Server für Programme der numerischen Mathematik ist NETLIB. Die größte entsprechende Sammlung in Deutschland ist die eLib des Konrad-Zuse-Zentrums in Berlin. Dort liegen neben einer Kopie der NETLIB auch zahlreiche aktuelle Entwicklungen wie die im Text erwähnten Routinen LIMEX und NLEQ1. Diese Sammlung ist über DATED-P, INTERNET und die verschiedenen Mail-Protokolle leicht erreichbar. Die Netzwerkverwalter von wissenschaftlichen Rechenzentren können in der Regel über den besten Zugang Auskunft geben.

Die mitgeteilten Erfahrungen entstammen zum Teil dem Gemeinschaftsprojekt DIVA zur dynamischen Simulation verfahrenstechnischer Anlagen von vier Instituten der Fakultät Verfahrenstechnik der Universität Stuttgart. Die Förderung dieser Arbeiten durch die Volkswagen-Stiftung und die Deutsche Forschungsgemeinschaft wird dankbar anerkannt.

Eingegangen am 21. November 1991 [B 5688]

## Formelzeichen

$A$	Koeffizientenmatrix
$b$	Vektor der rechten Seite
$B, B$	Kapazitätsterm
$c$	Konzentration
$D$	Diffusionskoeffizient
$f, f$	Funktion der rechten Seite
$J$	Jacobi-Matrix
$k_1, k_2$	Geschwindigkeitskonstanten
$L$	Linksdreiecksmatrix
$L$	Länge des Reaktors
$N$	Anzahl der Gleichungen
$NZ$	Anzahl der Ortsstützstellen
$p$	Druck
$R$	Rechtsdreiecksmatrix
$S$	Quellenfunktion
$t$	Zeit
$T$	Temperatur
$v$	Strömungsgeschwindigkeit
$y, y$	Unbekannte
$z$	Ort
$z$	algebraische Unbekannte (in Abschnitt 4.3)
$\Delta t$	Zeitschrittweite
$\Delta z$	Ortsschrittweite
$\tau$	Zeitkonstante
$\omega$	Relaxationsfaktor

Vektoren und Matrizen werden durch fetten Druck gekennzeichnet.

## Indices

$i$	Iterationsindex
$k$	Zeitindex oder Summationsindex
$l$	Ortsindex
*	Zulauf

## Literatur

- [1] *Stoer, J.*: Numerische Mathematik 1, Springer-Verlag, Berlin 1989, 5. Aufl.
- [2] *Stoer, J.; Bulirsch, R.*: Numerische Mathematik 2, Springer-Verlag, Berlin 1990, 3. Aufl.
- [3] *Deuffhard, P.; Hohmann, A.*: Numerische Mathematik, W. de Gruyter, Berlin 1991.
- [4] *Engeln-Müllges, G.*: Numerische Mathematik für Ingenieure, Bibliographisches Institut, Mannheim 1987, 5. Aufl.
- [5] *Schwarz, H. R.*: Numerische Mathematik, B. G. Teubner, Stuttgart 1988, 2. Aufl.
- [6] *Press, W. H., et al.*: Numerical Recipes, Cambridge University Press, Cambridge 1988, 3. Aufl.
- [7] *Burden, L. B.; Faires, J. D.*: Numerical Analysis, Prindle, Weber and Schmidt, Boston 1989, 3. Aufl.
- [8] *Gear, G. W.*: Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs 1971.
- [9] *Meis, T.; Marcowitz, U.*: Numerische Behandlung partieller Differentialgleichungen, Springer-Verlag, Berlin 1978.
- [10] *Ames, W. F.*: Numerical Methods for Partial Differential Equations, Academic, New York 1977, 2. Aufl.
- [11] *Davis, M. E.*: Numerical Methods and Modeling for Chemical Engineers, John Wiley and Sons, New York 1984.
- [12] *Finlayson, B. A.*: Nonlinear Analysis in Chemical Engineering, McGraw-Hill, New York 1980.

- [13] Patankar, S. V.: Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York 1980.
- [14] Richter, W.: Numerische Lösung partieller Differentialgleichungen mit der Finite-Elemente-Methode, F. Vieweg und Sohn, Braunschweig 1986.
- [15] Villadsen, J. V.; Michelsen, M. L.: Solution of Differential Equation Models by Polynomial Approximation, Prentice-Hall, Englewood Cliffs 1980.
- [16] Schwarz, H. R.: Methode der finiten Elemente, B. G. Teubner, Stuttgart 1984, 2. Aufl.
- [17] Deuflhard, P.: Numerik von Anfangswertmethoden für gewöhnliche Differentialgleichungen, FU Berlin 1989, TR 89-2.
- [18] Dongarra, J. J., et al.: LINPACK User's Guide. SIAM, Philadelphia 1979.
- [19] IMSL: User's Manual, IMSL Math/Library, Houston 1989.
- [20] Duff, I. S.; Reid, J. K.: ACM Trans. Math. Software 5 (1979) S. 18/35.
- [21] Duff, I. S.; Erisman, A. M.; Reid, J. K.: Direct Methods for Sparse Matrices. Clarendon Press, Oxford 1987.
- [22] Stadtherr, M. A.; Wood, E. S.: Comput. Chem. Eng. 8 (1984) S. 9/33.
- [23] Gilbert, J.; Peierls, T.: SIAM J. Sci. Stat. Comp. 9 (1988) S. 862/874.
- [24] Osterby, O.; Zlatev, Z.: Direct methods for sparse matrices Daimi PB-123, Aarhus University, Computer Science Dept., 1980.
- [25] Ng, E.: A comparison of some methods for solving sparse nonsymmetric linear systems, Oak Ridge National Laboratory 1991 (unpublished draft).
- [26] Saadi, Y.; Schultz, M. H.: GMRES: SIAM J. Sci. Stat. Comput. 7 (1986) S. 856/870.
- [27] Crowe, C. M.; Nishio, M.: AIChE J. 21 (1975) S. 3/12.
- [28] Broyden, C. G.: Math. Comp. 19 (1965) S. 577/593.
- [29] Schubert, L. K.: Math. Comp. 25 (1970) S. 27/39.
- [30] Bogle, I. D. L.; Perkins, J. D.: Comp. Chem. Eng. 12 (1988) S. 791/812.
- [31] Deuflhard, P.: Newton Techniques for Highly Nonlinear Problems – Theory, Algorithms, Codes. Academic press, New York (to appear).
- [32] Shacham, M.: Comp. Chem. Eng. 9 (1985) S. 103/124.
- [33] Sun, E. T.; Stadtherr, M. A.: Comp. Chem. Eng. 12 (1988) S. 1129/1145.
- [34] Stoer, J., in: Computational Mathematical Programming, Klaus Schittkowski, Hrsg., NATO ASI Series, Springer, Berlin 1985.
- [35] Char, B. W., et al.: MAPLE Reference Manual, Watcom Publ. Ltd., 1988.
- [36] MACSYMA Reference Manual: Symbolics Inc. 1988.
- [37] Deuflhard, P.: SIAM Rev. 27 (1988) S. 505/535.
- [38] Gear, C. W.: IEEE Trans. Circuit Theory CT-18 (1971) S. 89/95.
- [39] Pantelides, C. C.: SIAM J. Sci. Stat. Comput. 9 (1988) S. 213/231.
- [40] Petzold, L. R.: SIAM J. Sci. Stat. Comput. 3 (1982) S. 367/384.
- [41] Gear, C. W.; Petzold, L. R.: SIAM J. Numer. Anal. 21 (1984) S. 716/728.
- [42] Hindmarsh, A. C.; Johnson, S. H.: Chem. Eng. Sci. 43 (1988) S. 3235/3258.
- [43] Brenan, K. E.; Campbell, S. L.; Petzold, L. R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, North-Holland, Amsterdam 1989.
- [44] Hindmarsh, A. C.: ACM-Signum Newsletter 15 (1980) S. 10/27.
- [45] Bachmann, R.; Mrziglod, T., et al., in: Dechema-Monogr. 116, VCH, Weinheim 1989.
- [46] Bachmann, R.; Mrziglod, T., et al.: Comp. Chem. Eng. 14 (1990) S. 1271/1273.
- [47] Deuflhard, P.; Hairer, E.; Zugck, J.: Numer. Math. 51 (1987) S. 501/516.
- [48] Brenan, K. E.; Petzold, L. R.: SIAM J. Numer. Anal. 26 (1989) S. 976/996.
- [49] Ascher, U. M.; Petzold, L. R.: SIAM J. Numer. Anal. 28 (1991) S. 1097/1120.
- [50] Hairer, E.; Lubich, C.; Roche, M.: The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods, Springer, Berlin 1989.
- [51] Kröner, A., et al.: Comp. Chem. Eng. 14 (1990) S. 1289/1295.
- [52] Perkins, J. D.; Sargent, R. W. H.: AIChE Symp. Ser. 214 (1982) S. 1/11.

## Buchbesprechung

**Handbuch der industriellen Fest/Flüssig-Filtration.** Herausgeg. von H. Gasper. Hüthig Verlag, Heidelberg 1990. 338 S., 140 Abb., 13 Tab., geb., DM 98,-.

Das Handbuch der industriellen Fest/Flüssig-Filtration ist eine Sammlung von Beiträgen zahlreicher Fachleute auf dem Gebiet der Fest/Flüssig-Trennung. Der Herausgeber hat versucht, in die Aneinanderreihung der Kapitel eine gewisse Ordnung zu bringen, indem er sie in „Grundlagen“, „Filterbauarten“ und „Peripherie“ unterteilt hat. Im Hauptteil werden 18 Filterbauarten in alphabetischer Reihenfolge abgehandelt, von „A“ wie Anschwemmsysteme bis „T“ wie Trommelfilter, einschließlich der neueren Techniken der Membranfiltration (unter „M“) und der dynamischen Filtration (unter „D“). Im Vorspann werden allgemeine Überlegungen zu Entwicklungstendenzen angestellt und theoretische Grundlagen

erläutert sowie unter „Neuentwicklungen“ auch Druck-Filterzentrifugen vorgestellt. Im dritten Teil („Peripherie“) werden u.a. Filtermedien, Filterhilfsmittel, Test- und Auslegungsmethoden behandelt. Den Schluß bildet ein umfangreiches Bezugsquellen- und ein Stichwortverzeichnis.

Die unterschiedlichen Autoren haben zwangsläufig ein ziemlich heterogenes Buch entstehen lassen, wobei in manchen Kapiteln die Firmeninteressen einzelner Fachleute erkennbar sind. Viele der Beiträge sind jedoch ausgesprochen informativ und praxisnah, so daß das Buch in der Hand von Planern und Betreibern eine nützliche, kompakte Informationsquelle darstellt. Zumindest im deutschen Sprachraum gibt es zur Zeit keine vergleichbar aktuelle Zusammenstellung des Standes der Fest/Flüssig-Filtration.

[BB 2832] E. Schubert, Ludwigshafen/Rh.