Entwicklung einer Schnittstelle für einen DAE–Solver in der chemischen Verfahrenstechnik

Dietmar Horn *
Weierstraß–Institut für Angewandte Analysis und Stochastik
Mohrenstraße 39, D-10117 Berlin

1991 Mathematics Subject Classification. 68U30, 65C99.
Keywords. software engineering, interfaces.

* email: horn@wias-berlin.de

Zusammenfassung

Bei der Entwicklung numerischer Verfahren zur Lösung großer strukturierter Systeme von Algebro–Differentialgleichungen, die in der chemischen Prozeßsimulation vorkommen, ist eine möglichst automatische Bereitstellung der Beispieldaten notwendig. Dazu wurde eine Schnittstelle definiert, die die notwendigen Informationen enthält. Zur automatischen Schnittstellenerzeugung wurde ein Programm implementiert, das ausgehend von den Protokollen des Prozeßsimulators SPEEDUP diese Schnittstelle erzeugt.

Seit 1994 arbeitet eine Arbeitsgruppe im WIAS Berlin im Rahmen des BMBF–Förderprogramms "Anwendungsorientierte Verbundvorhaben auf dem Gebiet der Mathematik" an der Entwicklung von numerischen Verfahren zur Lösung großer Systeme von Algebro–Differentialgleichungen (DAE), die bei der Modellierung verfahrenstechnischer Prozesse in der chemischen Industrie auftreten (siehe [3]). Diese DAE–Systeme, die oft mehrere 10 000 Gleichungen umfassen, sind in der Regel in Teilsysteme (Funktionsblöcke) strukturiert.

Probleme dieser Größenordnung erfordern zur Formulierung der verfahrenstechnischen Prozesse die Verwendung angepaßter Beschreibungsmittel. Die Entwicklung einer Beschreibungssprache und des dazugehörigen Compilers ist auf Grund der Komplexität der zu beschreibenden Prozesse mit sehr großem Aufwand verbunden. Diese Arbeiten können im Rahmen der Entwicklung von numerischen Verfahren nicht erfolgen. Darüber hinaus würde die Akzeptanz einer Eigenentwicklung bei den Anwendern in den Universitäten, Forschungseinrichtungen und in der Industrie nur sehr begrenzt sein, da bereits mehrere chemische Prozeßsimulatoren mit entsprechenden Eingabemöglichkeiten existieren und genutzt werden. Für unsere Arbeiten war es also notwendig, einen vorhandenen, möglichst weit verbreiteten Simulator zu benutzen und eine Übernahme der für die numerischen Verfahren wichtigen Informationen zu realisieren.

Von den in Frage kommenden Simulatoren stellt keiner eine vollständig beschriebene Schnittstelle zu den numerischen Verfahren zur Verfügung, die alle von uns benötigten Informationen enthält. Eine einfache Ersetzung der benutzten numerischen Verfahren war deshalb bei keinem Simulator möglich. Es mußte also zuerst, ausgehend von den zu lösenden DAE–Systemen und den numerischen Verfahren, eine Schnittstelle definiert werden, die die notwendigen Informationen enthält. Dann mußte ein Simulator gesucht werden, der es erlaubte, diese Informationen zu erzeugen. Danach konnte man die Schnittstelle durch weitere durch diesen Simulator erreichbare Informationen erweitern.

1 Schnittstellendefinition

Die mathematische Modellierung verfahrenstechnischer Prozesse in chemischen Anlagen, etwa bei der dynamischen Simulation komplexer chemischer und physikalischer Vorgänge in wärme– und stromverkoppelten Destillationskolonnen, führt auf Anfangswertprobleme für große Systeme von Algebro–Differentialgleichungen

$$F(t, y(t), \dot{y}(t), u(t)) = 0, \;\; y(t_0) = y_0,$$

$$F : \mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}^n, t \in [t_0, t_{END}],$$

wobei die Parameterfunktion $u(t)$ gegeben und $y(t) = (x_1(t), ..., x_n(t))^T$ gesucht ist. Die Systeme können mehrere 10 000 Gleichungen umfassen und sind entsprechend der Modellierung der Gesamtanlage nach Funktionsblöcken in Teilsysteme strukturiert. Die Schnittstellendefinition wurde hauptsächlich durch die für die Berechnung der DAE–Systeme notwendigen Daten bestimmt. Eine wesentliche Besonderheit bestand in der Beschreibung der Struktur des DAE–Systems.

Die Schnittstelle mußte folgende Bestandteile umfassen:

- Beschreibung der Variablen des DAE–Systems (Vektoren $y(t)$ und $\dot{y}(t)$)
- Anfangswerte für die Variablen
- Anfangswerte für die Parameter
- Beschreibung der rechten Seiten der Teilsysteme des DAE–Systems (Vektor F)
- Beschreibung der schwachbestzten Jacobi–Matrix des DAE–Systems (Matrix A)
- Subroutinen zur teilsystemorientierten Berechnung des Vektors F und der Matrix A

Es gehören also nicht nur die Beschreibungsdaten und eventuelle Anfangswerte zur Schnittstelle, sondern auch Subroutinen, die notwendige Berechnungen durchführen. Wir haben dementsprechend eine Datenschnittstelle und zwei Programmschnittstellen definiert (siehe [1]).

1.1 Datenschnittstelle

- Angaben zur Größe des DAE–Systems:
  Anzahl der Gleichungen, Teilsysteme, dynamischen Variablen $\dot{y}$, Parameter u und Nicht–Null–Elemente der Jacobi–Matrix A
- Angaben zur Struktur des DAE–Systems mit der Zuordung der Gleichungen zu den Teilsystemen
- Angabe der y–Indizes der dynamischen Variablen
- Angabe der Struktur der schwachbesetzten Jacobi–Matrix
- Anfangswerte der Variablen y, dynamischen Variablen $\dot{y}$ und Parameter u

1.2 Programmschnittstelle

- Subroutinen–Aufruf zur teilsystemorientierten Berechnung der Funktionswerte:
  FUNCTION(k,t,Y,DY,F),
  wobei k das Teilsystem identifiziert, t der Integrationszeit, Y dem Vektor $y(t)$, DY dem Vektor $\dot{y}(t)$ und F dem Vektor F entsprechen.

- Subroutinen–Aufruf zur teilsystemorientierten Berechnung der Jacobi–Matrix:
  JACOBIAN(k,t,c,Y,DY,F,A),
  wobei zusätzlich c eine Integrationskonstante und A der Vektor der Nicht–Null–Elemente der Jacobi–Matrix A sind.

2 Schnittstellenerzeugung

Eine Untersuchung der verfügbaren Simulatoren ergab, daß der kommerzielle chemische Prozeßsimulator SPEEDUP (Aspen Technology, Inc., Cambridge, Massachusetts, USA, [4], [5]) alle von uns geforderten Voraussetzungen erfüllt. Er erlaubt eine strukturierte Beschreibung der chemischen Prozesse, und seine Protokolle enthalten alle notwendigen Informationen.

Nachteilig war nur, daß die Strukturierung durch den Simulator aufgelöst wird und deshalb auch die Routinen zur Berechnung der Funktionswerte und der Elemente der Jacobi–Matrix eine teilsystemorientierte Berechnung nicht vorsehen.

Die ursprüngliche Strukturierung des DAE–System mußte also wieder rückerkannt werden. Aus diesem Grund und um auch große praxisrelevante Beispiele bearbeiten zu können, wurde ein Programm entwickelt, das automatisch die von uns definierte Schnittstelle erzeugt. Die Rückerkennung der Teilsystemstruktur erfolgt ebenfalls überwiegend automatisch. In seltenen Fällen ist ein Dialog zur Angabe nicht erkannter Teilsysteme notwendig. Automatisch werden auch die Subroutinen FUNCTION und JACOBIAN erzeugt, die mit Hilfe der von SPEEDUP erzeugten Routinen die teilsystemorientierte Berechnung erlauben.

Es wurden bisher Beispiele bis zu einer Größe von etwa 14 000 Gleichungen bearbeitet und von den numerischen Verfahren verwendet.

2.1 SPEEDUP–Beschreibungssprache

Am Beispiel eines Doppel–Effekt–Verdampfers mit 13 Teilsystemen (Bild 1) soll die SPEEDUP–Beschreibungssprache kurz vorgestellt werden. Sie unterstützt eine gleichungsorientierte Modellierung chemischer Prozesse.

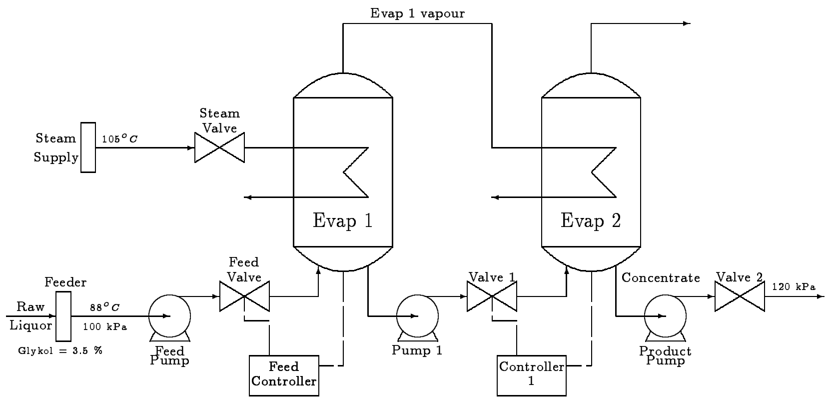

Bild 1: DYNEVAP

Eine Strukturierung wird durch die Definition von UNITs erreicht.

```
UNIT   EVAP1        is a   EVAPORATOR
UNIT   EVAP2        is a   EVAPORATOR
UNIT   FEED_PUMP    is a   PUMP
UNIT   PROD_PUMP    is a   PUMP
UNIT   PUMP1        is a   PUMP
UNIT   VALVE1       is a   VALVE
UNIT   VALVE2       is a   VALVE
```

Die Beschreibung der Einzelprozesse erfolgt modular durch das MODEL–Sprachelement.

```
MODEL PUMP
 Set NOCOMP
 Type
   FLOW,FLOW1,LEAK               as   ARRAY(NOCOMP) of FLOWRATE
   TOTFLOW,MAXFLOW,TOTLEAK       as   FLOWRATE
   PRESS_IN,PRESS_OUT            as   PRESSURE
   DELP,DELPMAX,DELPMIN          as   PRESS_DROP
   ENTH,ENTH1                    as   LIQENTH
 Stream
   Input 1    is   FLOW,ENTH,PRESS_IN
   Output 1   is   FLOW1,ENTH1,PRESS_OUT
 Equation
   DELP                 =   PRESS_OUT - PRESS_IN;
   TOTFLOW              =   SIGMA(FLOW);
   FLOW1 + LEAK         =   FLOW;
   ENTH                 =   ENTH1;
   TOTLEAK              =   SIGMA(LEAK);
   LEAK(1) * TOTFLOW    =   FLOW(1) * TOTLEAK;
```

Der Gesamtprozeß wird mit Hilfe der UNITs in dem FLOWSHEET–Teil beschrieben.

```
FLOWSHEET
    Output    of FEEDER       is Input      of FEED_PUMP
    Output    of FEED_PUMP    is Input      of FEEDVALVE
    Output    of FEEDVALVE    is Input 1    of EVAP1
    Output 1  of EVAP1        is Input      of PUMP1
    Output 2  of EVAP1        is Input 2    of EVAP2
    Output    of PUMP1        is Input      of VALVE1
    Output    of VALVE1       is Input 1    of EVAP2
    Output 1  of EVAP2        is Input      of PROD_PUMP
    Output    of PROD_PUMP    is Input      of VALVE2
    Output    of VALVE2       is Product 1
```

Den Größen in der SPEEDUP–Beschreibung können in einem DECLARE–Teil typbezogene Initialisierungsgrößen und Grenzen zugeordnet werden. Diese Angaben ordnen jeder Variablen des DAE–Systems einen Initialisierungswert und Grenzen zu. Diese Informationen wurden zusätzlich in unsere Schnittstellendefinition aufgenommen, da sie für die numerischen Verfahren wichtige Zusatzinformationen enthalten.

```
DECLARE
 Type
    FLOWRATE =  80 :      0 : 10000 UNIT = "kg/min"
    LIQENTH  = 100 :      0 :   400 UNIT = "kJ/kg"
    NOTYPE   =   1 : -1E35 :  1E35 UNIT = " - "
```

2.2 SPEEDUP–Dateien

Ein Simulationslauf mit SPEEDUP erzeugt alle für die Schnittstellenerzeugung notwendigen Informationen in folgenden Dateien:

- Beschreibungstext
  Neben dem SPEEDUP–Beschreibungstext werden hier durch den Befehle SAVE Informationen zu den berechneten Resultaten abgespeichert. Diese Resultate enthalten neben den berechneten Werten auch eine Liste der Teilsysteme und SPEEDUP–Größen.

- Protokoll
  Bei der Benutzung eines hohen Print–Levels bei der Simulation sind hier Listen der SPEEDUP–Größen mit ihrer internen Numerierung abgespeichert. Weiter enthält es auch Informationen zu den Nicht–Null–Elementen der Jacobi–Matrix. Die numerisch zu differenzierenden Matrix–Elemente sind markiert.

- FORTRAN–Programme
  Es werden Programme erzeugt, die zur Berechnung der Funktionswerte und der Jacobi–Matrix dienen. Ein weiteres Programm enthält eventuell im Beschreibungstext benutzte zusätzliche Prozeduren und Funktionen.

Das von uns entwickelte Programm analysiert diese Dateien und erzeugt die beschriebene Schnittstelle. Dabei wird die Strukturierung des Problems rückerkannt.

3 Schlußbemerkungen

Unser Forschungsthema war ohne Bereitstellung einer Schnittstelle zu den zu entwickelnden numerischen Verfahren nicht zu bearbeiten. Vorhandene praxisrelevante Beispiele lagen nicht in einer verwendbaren Form vor. Es ergab sich also die Notwendigkeit, selbst für die Bereitstellung von Beispielen zu sorgen. In der beschriebenen Art war es möglich, das Problem ohne zu großen Aufwand zu lösen.

Der Entwurf der Schnittstelle war wesentlich durch das Problem (DAE–System) und die Anforderungen der numerischen Verfahren bestimmt. Es mußte eine Beschreibungsmöglichkeit für das Problem gesucht werden. Entsprechende Simulatoren besitzen alle eine Eingabemöglichkeit. Da in diesen Simulatoren ähnliche numerische Verfahren eingesetzt werden, müssen auch alle notwendigen Informationen verfügbar sein. Wenn ein solcher Simulator seine internen Daten in Form einer oder mehrerer Schnittstellen explizit verfügbar machen würde , könnte man alle Informationen erhalten, und auch der Simulator wäre leichter modifizierbar. Wir haben in SPEEDUP einen Simulator gefunden, der über eine ausführliche Protokollierung verfügt und uns dadurch eine Schnittstellenerzeugung möglich macht.

Erstrebenswert ist die Realisierung einer Schnittstelle, die sämtliche aus der Beschreibungssprache analysierten Informationen enthält, womit eine weitgehende Trennung des Simulators von der Beschreibungssprache erreicht wird. Damit ergäbe sich außerdem die Möglichkeit der Standardisierung der Beschreibungssprache für ein Anwendungsgebiet, wie z.B. die chemische Prozeßsimulation.

Auf dem Gebiet der Analyse elektrischer Netzwerke wurde von uns dieser Weg eingeschlagen. Für den Simulator MAGNUS (siehe [2]) wurde für die in diesem Bereich weitverbreitete Beschreibungssprache SPICE ein Compiler entwickelt, der eine definierte Schnittstelle erzeugt. Simulator und Compiler sind nur über diese Schnittstelle miteinander verbunden.

Literatur

[1] J. Borchardt, F. Grund, D. Horn, T. Michael, H. Sandmann, *Beschreibung der Schnittstelle eines Solvers für strukturierte DAE–Systeme auf MPP–Rechnern*, Weierstraß–Institut für Angewandte Analysis und Stochastik, Berlin, 1994.

[2] J. Borchardt, F. Grund, D. Horn, M. Uhle, *MAGNUS – Mehrstufige Analyse großer Netzwerke und Systeme*, Weierstraß–Institut für Angewandte Analysis und Stochastik, Report No. 9, Berlin, 1994.

[3] F. Grund, J. Borchardt, D. Horn, T. Michael, H. Sandmann, *Differential–algebraic systems in the chemical process simulation*, Proceedings Workshop Scientific Computing in Chemical Engineering, Hamburg, Springer Verlag, 1995 (eingereicht).

[4] J.D. Perkins, R.W.H. Sargent, *SPEEDUP: A Computer Program for Steady State and Dynamic Simulation and Design of Chemical Processes*, AIChE Symp. Ser. **78** (1982), 1 – 11.

[5] Aspen Technology, *SPEEDUP, User Manual, Library Manual*, Aspen Technology, Inc., Cambridge, Massachusetts, USA, 1995.