Theorie und Numerik von Differentialgleichungen
mit
MATLAB und SIMULINK

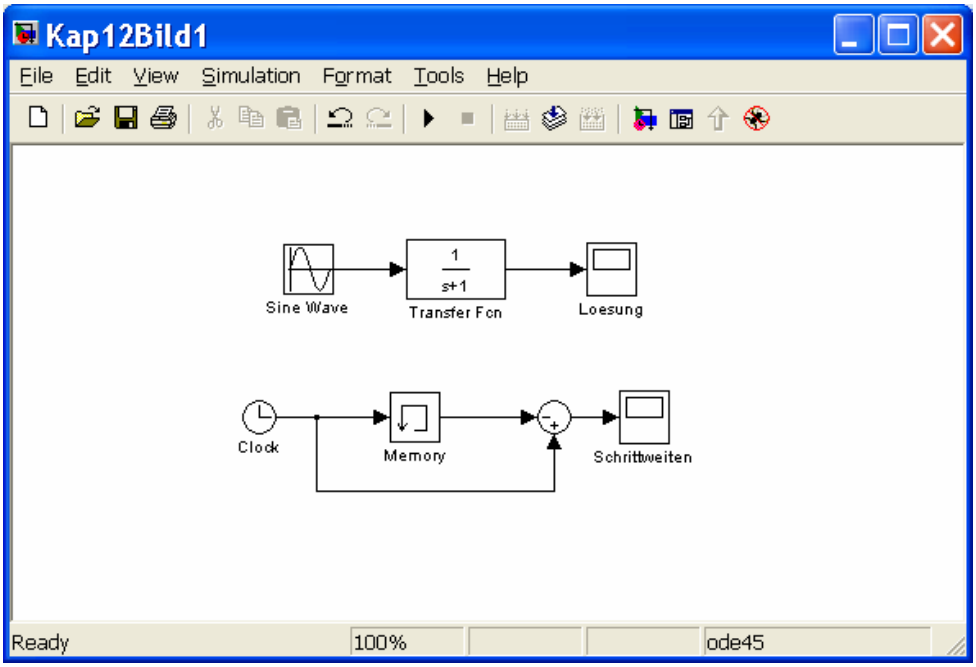
K. Taubert
Universität Hamburg
SS08

SIMULINK II

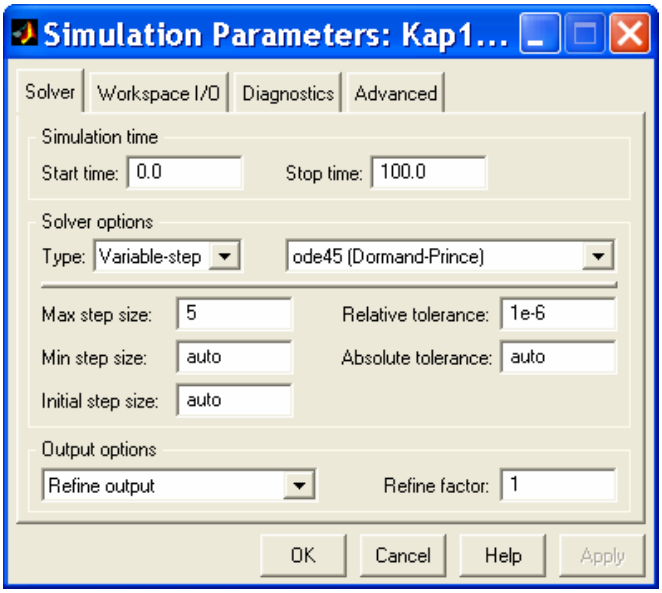
12 SUBSYSTEME IN SIMULINK

Subsysteme sind das Äquivalent zu den Unterprogrammen der üblichen Programmier-Sprachen. Große Modelle kann man durch Verwendung von Subsystemen vereinfachen und überschaubar machen.
Subsysteme lassen sich auch in anderen Modellen, in denen dieselbe Funktionalität erforderlich ist einsetzen und ermöglichen so die Wiederverwendbarkeit einer einmal gemachten Arbeit. Zu diesem Zwecke können Anwender eigene Block-Bibliotheken entwickeln

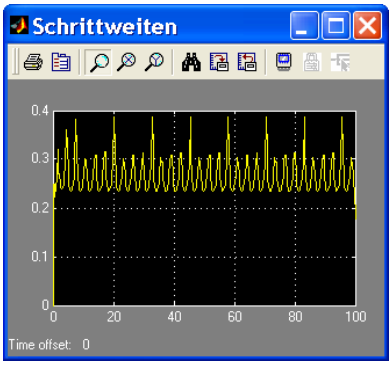
§ 12.1 Subsysteme und Masken



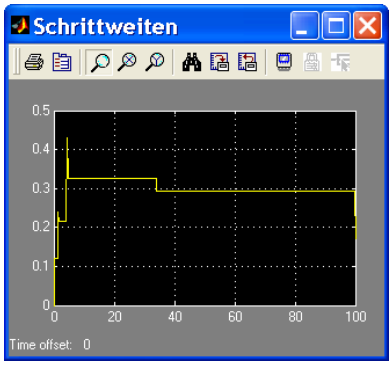
Das obige SIMULINK-Modell liefert neben der (Näherungs-) Lösung von y'+ y=sin(t), y(0)= 0, auch die Schrittweiten bei der Integration der Anfangswertaufgabe. Die zugehörigen Parameter können aus dem folgenden Fenster entnommen werden.



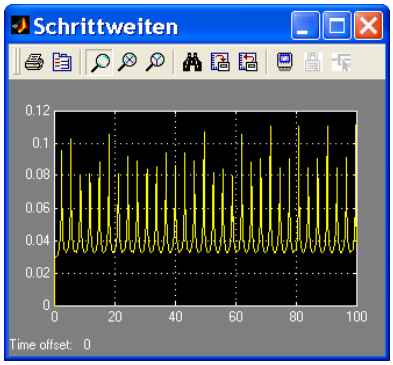
Die nachfolgenden Bilder zeigen die Schrittweiten für unterschiedliche Integratoren



ODE45(Dormand-Prince)

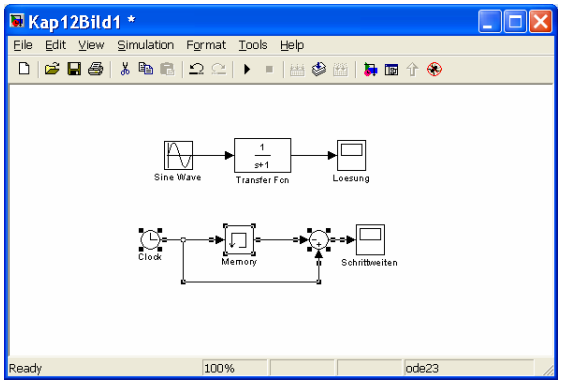


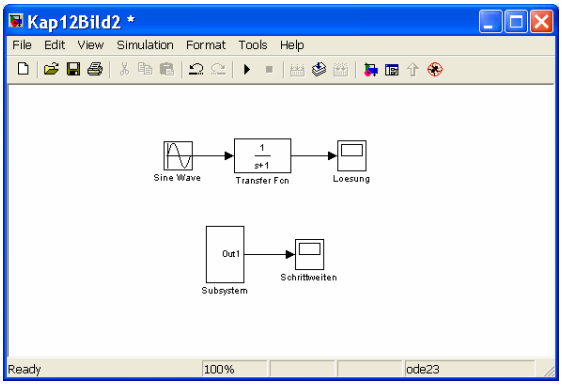
ODE113(Adams)



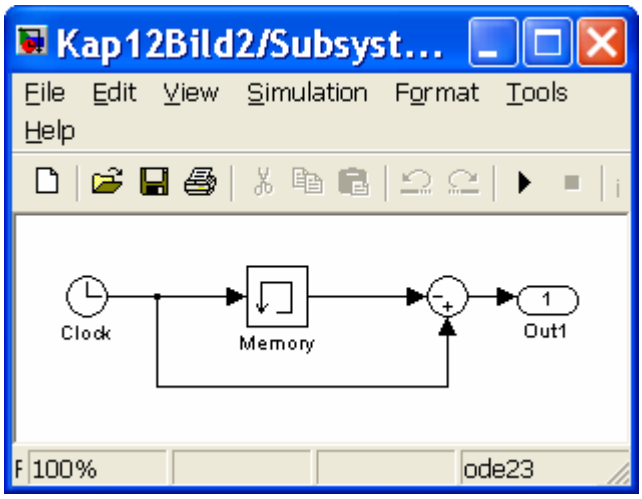
ODE23(Bobacki-Shampine)

Ist ein SIMULINK-Modell gegeben, dann können Teile des Modells als Subsysteme zusammengefasst werden. Dieses soll mit einem Teil der Blöcke zur Bestimmung der Schrittweiten erfolgen.
Markieren der Blöcke und die anschließende Funktion **Edit->Create Subsystem** liefert





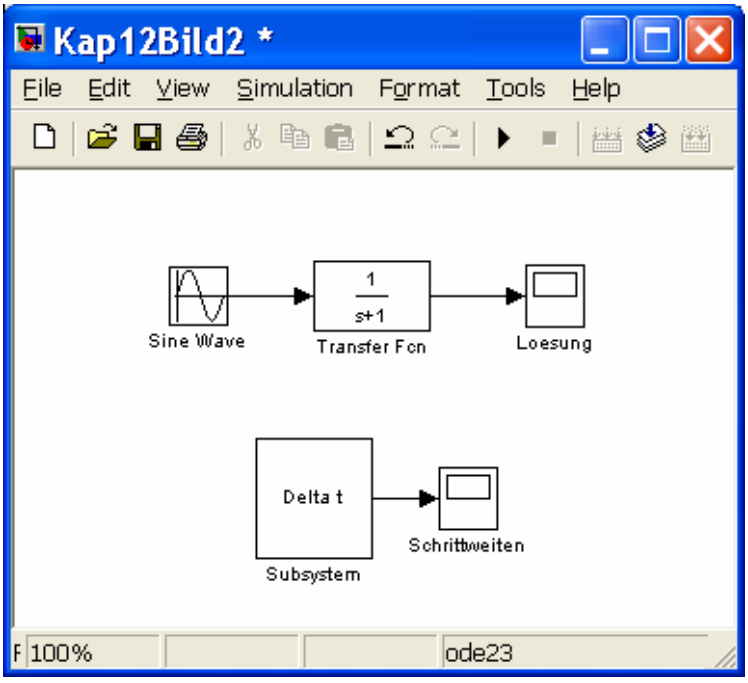
mit einen neuen Subsystem-Block. Anschließendes Anklicken des Subsystems oder **Look under Mask** gestattet einen Blick unter die Maske mit einem neuen Block „Out1“



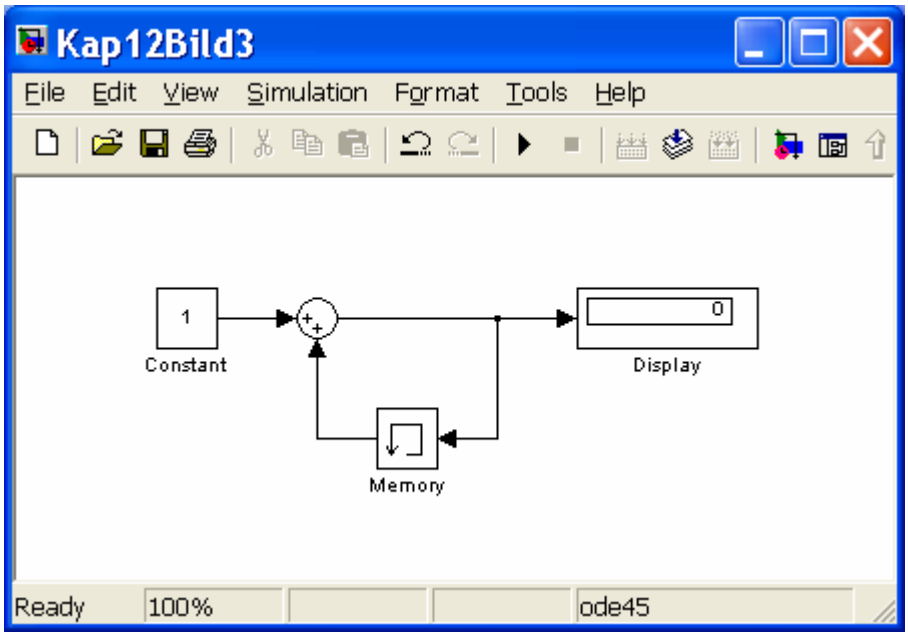
Eine Unbenennung vom Subsystemblock kann wie folgt erfolgen:

1. Subsystemblock anklicken.
2. **Edit-> Mask Subsystem.**
3. Im daraufhin erscheinenden Mask Editor Fenster geben wir im **Drawing commands** disp(´Delta t´) ein.
4. Nach anklicken von **Apply** können wir die Auswirkung auf den Icon des Subsystems erkennen.
5. Durch anklicken von **OK** wird das Fenster des Mask Editors geschlossen und dieser selbst beendet.

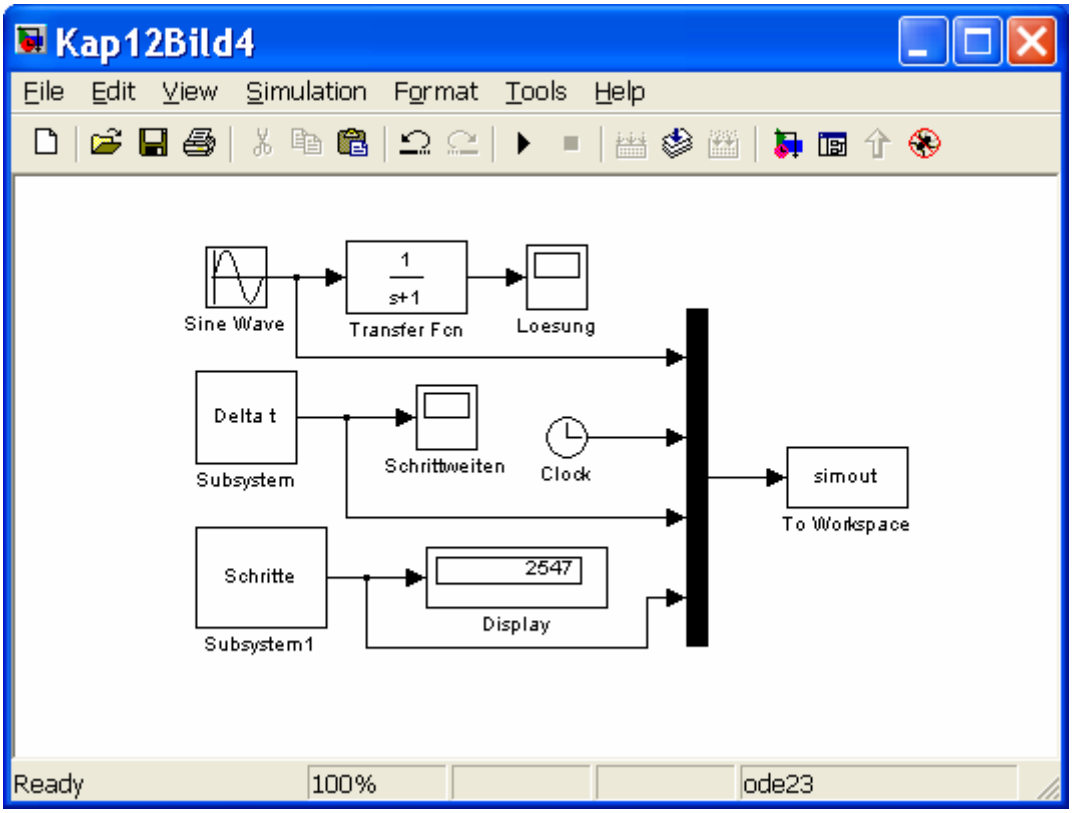
Das gesamte Modell mit dem Subsystem hat jetzt die Gestalt



Mit dem Modell

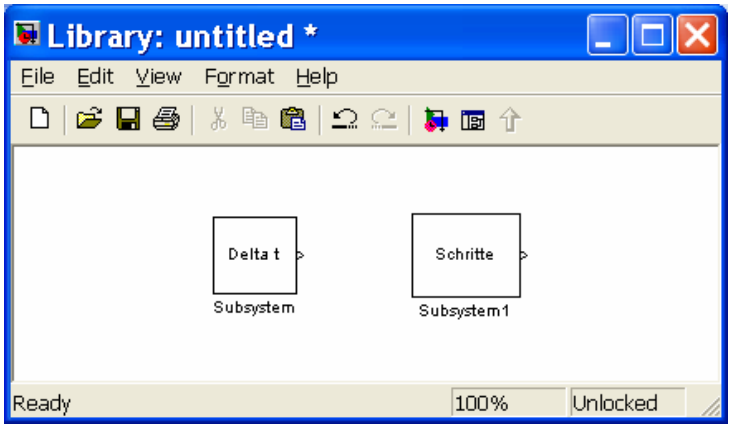


können die Integrationsschritte gezählt werden. Werden deren Blöcke (bis auf den Display Block) wie oben zu einen Subsystem mit den Namen Schritte zusammengefasst, dann kommen wir zu einen Modell mit dem die Lösung einer Anfangswertaufgabe, die verwendeten Schrittweiten, die Anzahl der Schritte und die verlaufende Zeit bestimmt und auf der MATLAB-Ebene ausgegeben werden kann.
Die Ausgabe auf der MATLAB-Ebene ist allerdings nur dann vollständig, wenn für die **Block-Parameter** von **To Workspace** als **Save Format** auch **Array** eingegeben ist



Beachte:
Durch „**Look under Mask**“ kann das Subsystem angesehen werden. „Create Subsystem“ kann aber nicht rückgängig gemacht werden.

Für die meisten Zwecke wird es genügen, unsere Subsysteme in ein eigenes Modell aufzubewahren, aus dem wir Sie nach Bedarf in andere Modelle kopieren.
Es kann aber auch eine neue Bibliothek angelegt werden. Dieses kann wie folgt erfolgen:
Im Fenster zum Modellieren wählen wir **File->New->Library** worauf ein mit ‚Library: untitled’ betiteltes Fenster erscheint und wir kopieren unsere Blöcke in dieses Fenster:



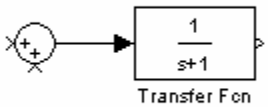
Mit **File->Save** im Bibliotheks-Fenster sichern wir die Bibliothek, wobei wir festlegen in welchem Verzeichnis sie unter welchem Namen zu erreichen ist.

Unabhängig von gegebenen Teilen eines SIMULIK-Modells können auch ad hoc neue Subsysteme erzeugt werden:

1. In Ports&Subsystems gehen.
2. Subsystem Box auswählen.
3. Doppelklicken auf Subsystems.
4. Aufbau des Subsystems
5. Anschließend unter geeigneten Namen speichern

Als Beispiel wollen wir ein Proportionalglied als Subsystem einführen und gleichzeitig die Möglichkeit eröffnen die auftretenden Parameter G =1/R und C vorzugeben:

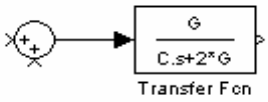
1. Mittels **File->New>Library** wird ein Bibliothek-Fenster geöffnet.
2. In dieses Fenster kopieren wir einen Summen-und einen Transfer-Block



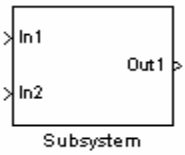
3. Als Parameter für den Transferblock geben wir ein

Numerator: [G]
Denominator: [C 2*G]

Der Icon des Transferblocks ändert sich daraufhin zu

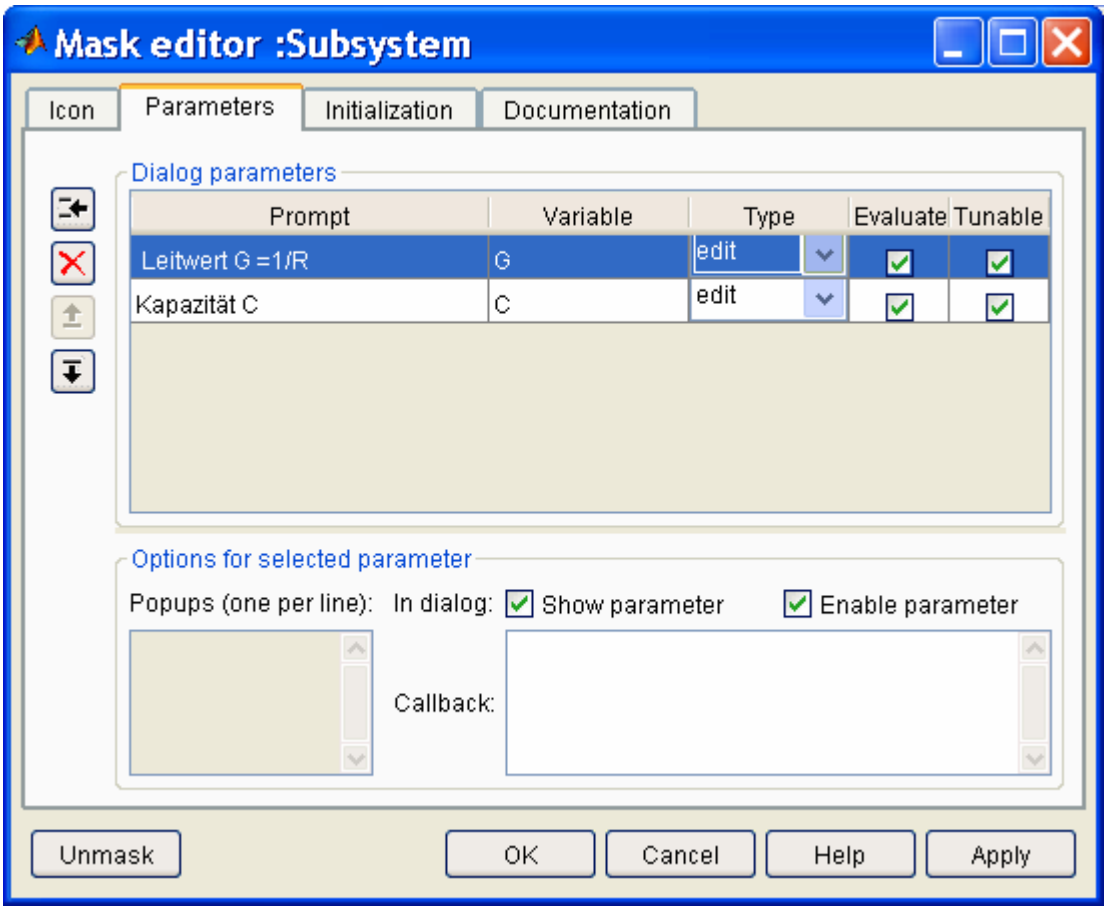


3. Wir markieren das gesamte Teilsystem und verwandeln es mittels **File->Create Subsystem** in ein Subsystem

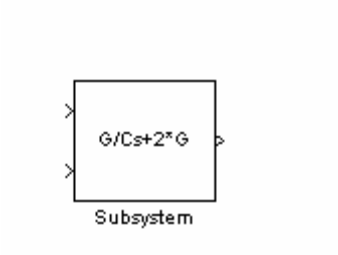


4. Wir markieren den Subsystem-Block und öffnen mittels **File->Mask Subsystem** den Maskeneditor. Eintrag von disp('G/Cs+2*G') (siehe z.B. **Examples of drawing commands)** liefert nach **Apply** später das Icon mit der Aufschrift (G/Cs+2*G)**.** Unter dem Register **Parameters** liefert anklicken von

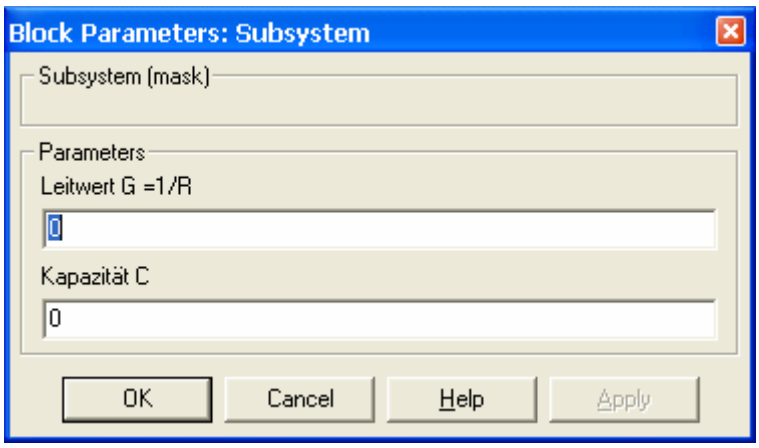
die Schreibflächen für die Prompts (Leitwert G =1/R, Kapazität C) und den Variablen (G,C) des Icons.



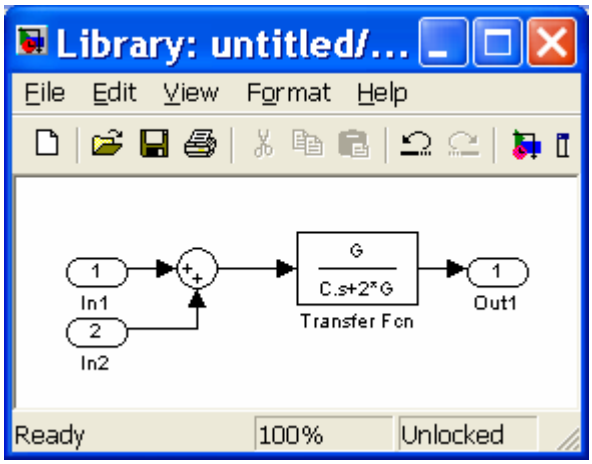
Die Aktion wird mit **Apply** abgeschlossen und mit **OK** verlässt man den Maskeneditor.
Damit entsteht ein Subsystem der Gestalt



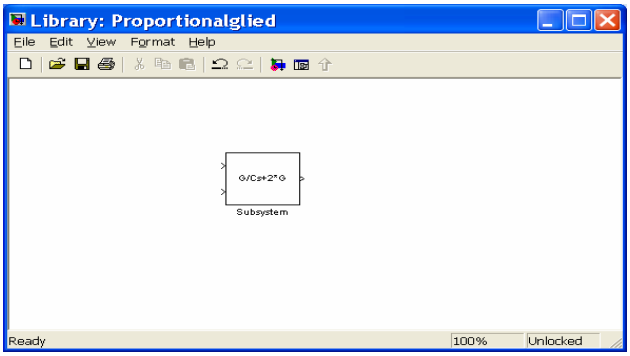
Wird der Subsystemblock angesprochen, erscheint das Fenster



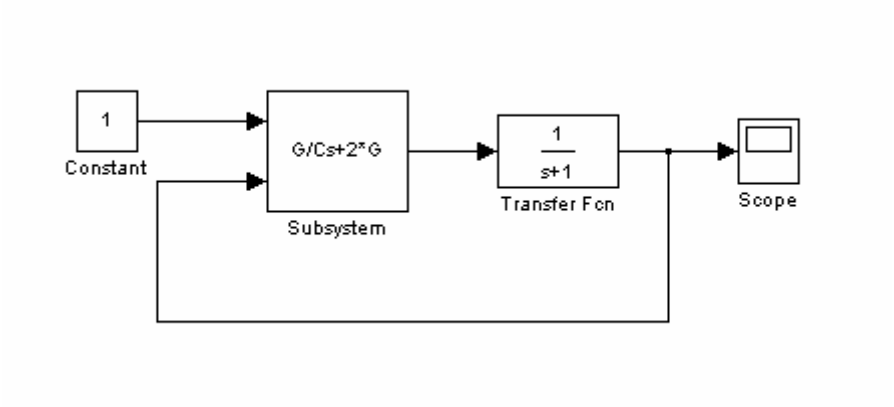
und die benötigten Parameter können eingetragen werden. Ein Blick unter den maskierten Block (Rechtsklick und **Look under Mask)** liefert



Wir sichern das Subsystem unter dem Titel Proportionalglied.



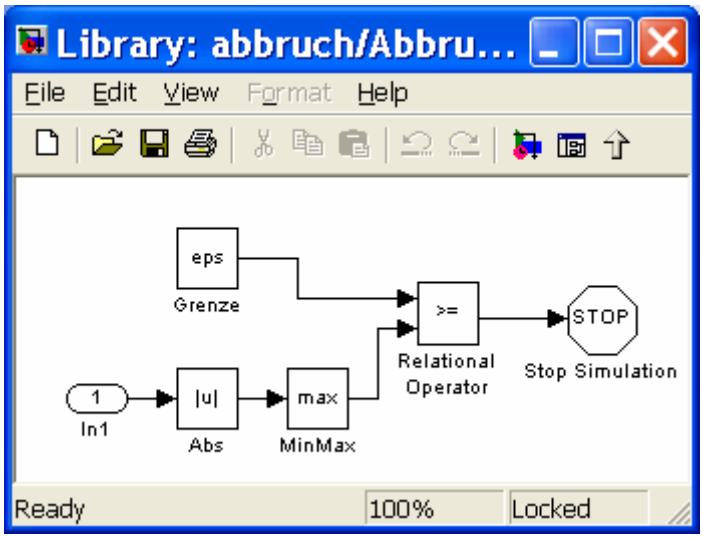
Ein SIMULINK-Modell mit dem so erzeugten Subsystem ist dann z.B.



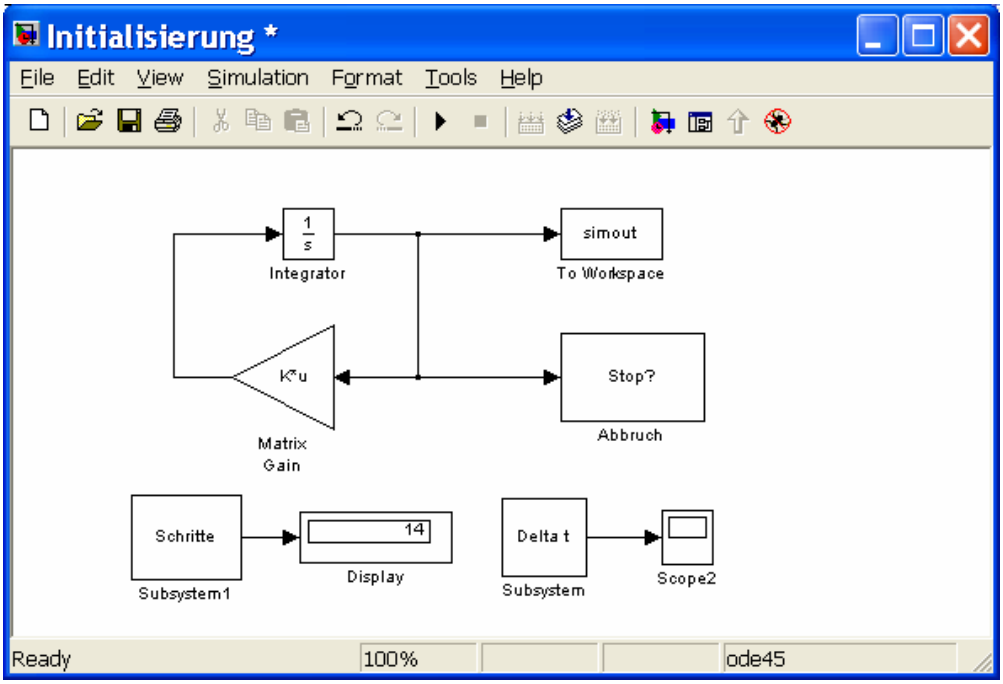
§ 12.2 Subsysteme: Initialisierung

Wir betrachten ein Problem mit einer vorgegebenen Anzahl n von Anfangswertaufgaben mit linearen Differentialgleichungen $y' = My$, $y(0) = y_0$ (z.B. Aufgabe 12.2). Die Anwender wählen eines davon aus

Mit dem folgenden Subsystem als Abbruchkriterium

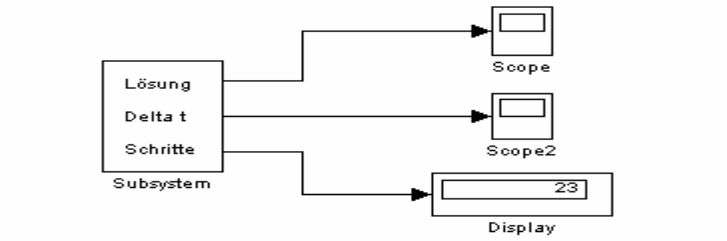


kann das folgende SIMULINK-Modell aufgebaut werden.

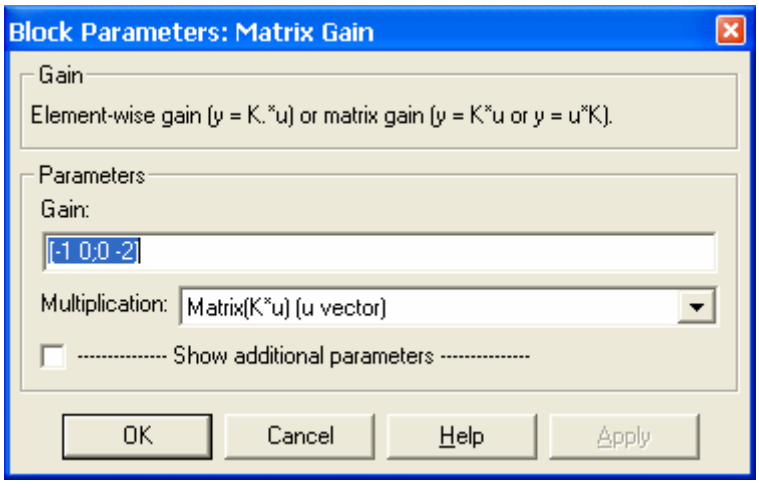


wobei im Matrix Gain Block die konkret ausgewählte Matrix M steht und im Integrator Block die vorgegebene Anfangsbedingung steht.

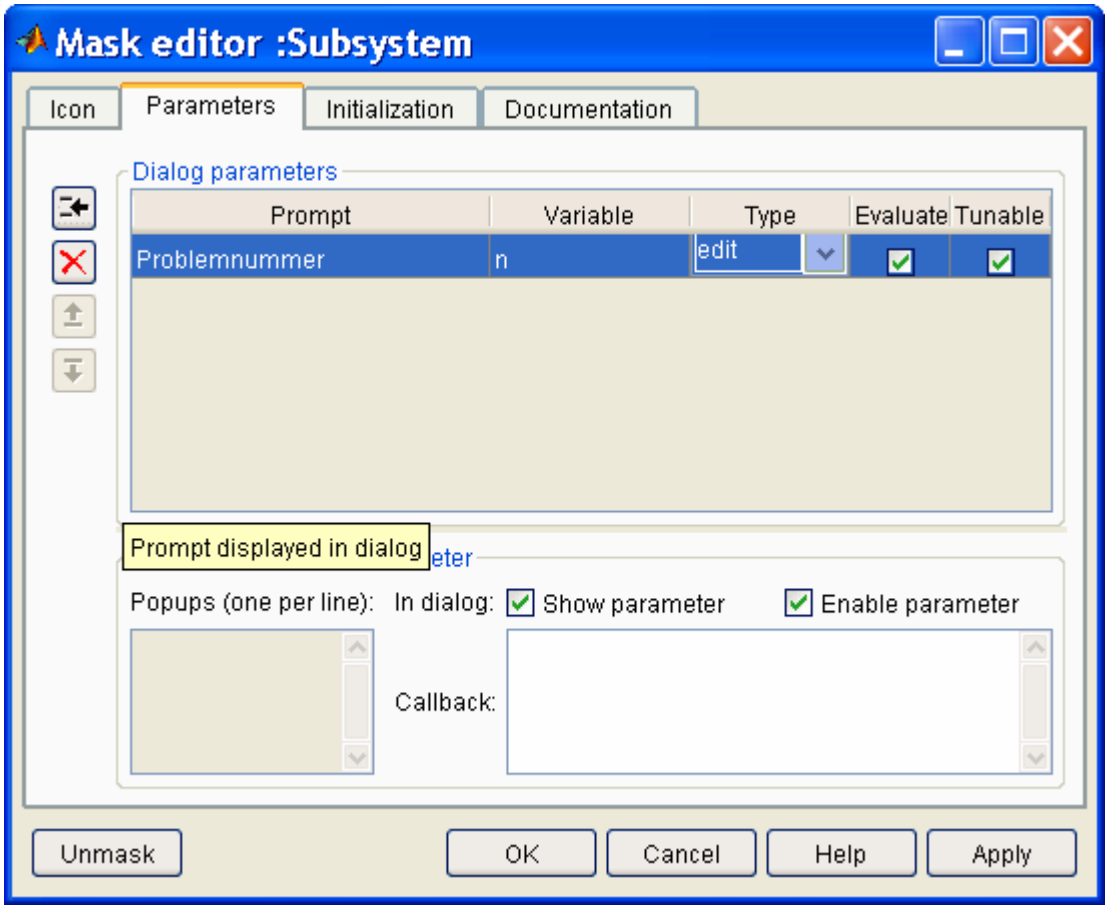
Es kann jetzt allgemein zweckmäßig sein das gesamte System, bis auf die Ausgabe-Blöcke, als *Subsystem* zu schreiben



Lästig ist nun, dass man im Parameterfenster des Subsystems für die Integration der Anfangswertaufgabe immer die ganze Matrix M eingeben muss.

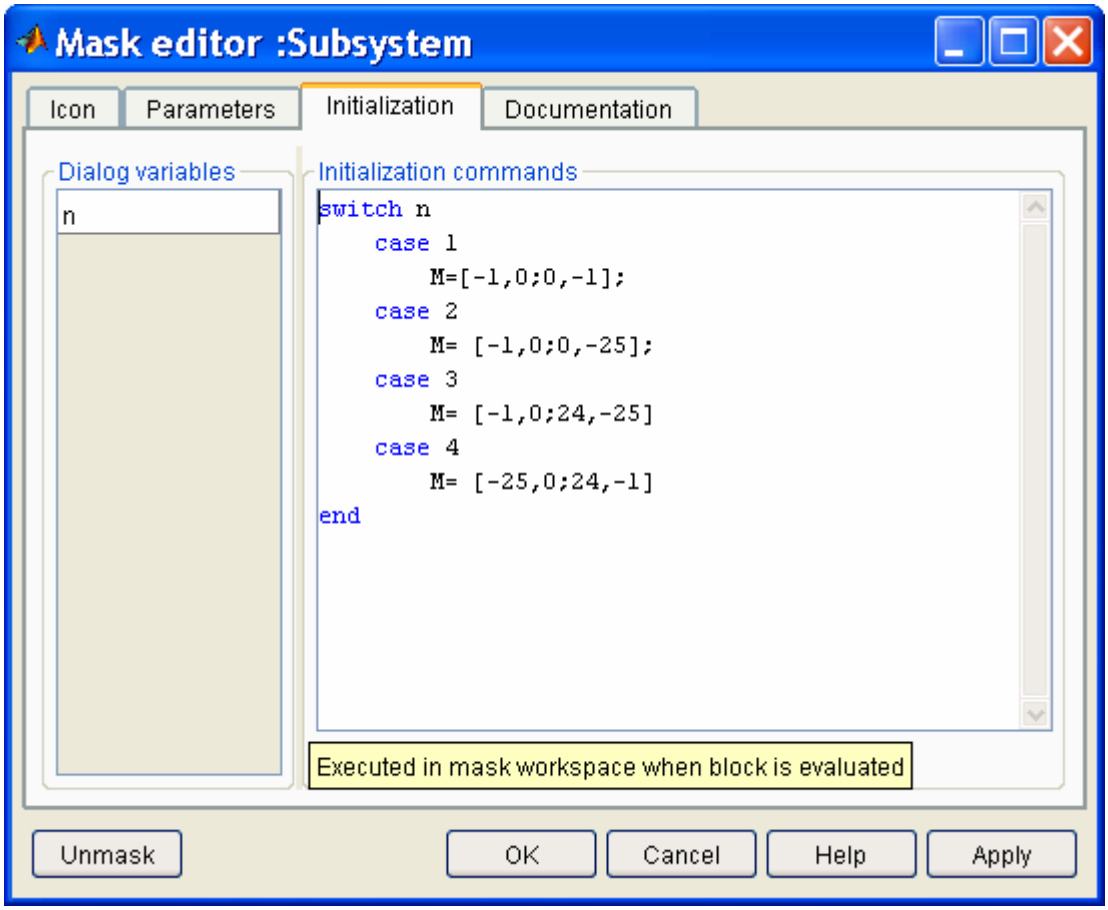


Dieses kann mühsam und fehleranfällig sein.
Bequemer kann die Verwendung von ‚Initializations commands' des *Subsystem* Blocks sein.
Wir tragen in der Maske zum Block-Parameter Matrix-Gain M ein.
Wir vereinbaren in der *Subsystem*-Maske (**Edit Mask**) unter **Dialog Parameters** eine Variable n mit dem Prompt Problemnummer.

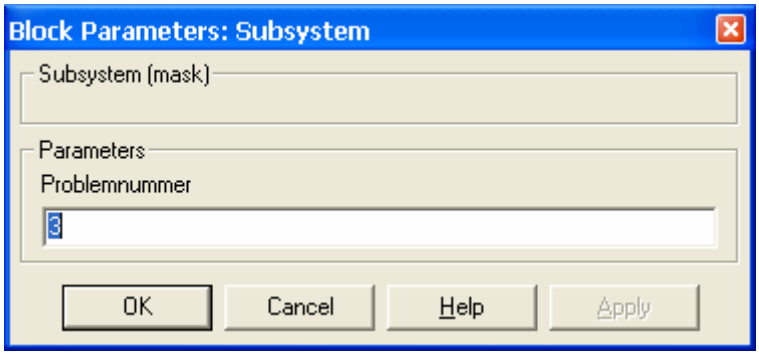


Die möglichen Variablenwerte sind 1,2,3 und 4 und sollen den Benutzer in einem Popup-Menue angeboten werden. Die Zuordnung zwischen Problemnummer und Matrix legen wir unter dem Register Initialization in den ‚Initialization commands' fest. Dieses wird durch MATLAB- Kommandos im Feld ‚Initialization commands' festgelegt. Wir geben also das folgende MATLAB-Programm ein:

```
switch n
case 1
M = [-1,0;0,-1];
case 2
M = [-1,0;0,25];
case 3
M = [-1,0;24,-25];
case 4
M = [-25,0;24,-1];
end.
```



Der Aufruf des *Subsystems* führt zum Menue



in den die gewünschte Problemnummer eingegeben werden kann.

Bemerkung
Mit dem bisherigen können Sie sich schnell ein (SIMULINK-)Modell zum Aufbau einer Statistik für verschiedene Integratoren aufbauen.

Literatur

[1] K.Taubert, W. Wiedl Differentialgleichungen mit SIMULINK. Vorlesung an der Universität Hamburg . WS99/00
[2] J.B. Dabney. T.L.Harman. Mastering SIMULINK 2. The Matlab Curriculum Series 1998