

Fast Solution for Large-Scale 2-D Convection-Diffusion, Reacting Flows

Hoang Duc Minh¹, Hans Georg Bock², Steffen Tischer¹, and Olaf Deutschmann¹ *

¹ Institute for Chemical Technology and Polymer Chemistry
University of Karlsruhe (TH)
Engesserstr. 20, D-76131 Karlsruhe, GERMANY
hoang.minh@kit.edu

² Interdisciplinary Center for Scientific Computing (IWR)
University of Heidelberg, GERMANY

Abstract. 2-D convection-diffusion, reacting flows in a single channel of catalytic monoliths are investigated. The fluid dynamics are modelled by a steady state, boundary-layer equations, which is a large system of parabolic partial differential equations (PDEs) with nonlinear boundary conditions arising from the coupling between the gas-phase and surface processes. The chemical processes are modelled using detailed chemistry. The PDEs are semi-discretized by a method of lines leading to a large-scale, structured differential algebraic equations (DAEs). The DAEs are solved using a tailored BDF code. We exploit the structure of the Jacobian and freeze the diffusion coefficients during approximation of Jacobian by the finite difference. By applying our approach, the computation times have been reduced by a factor of 4 to 10 and more depending on the particular problem.

1 Introduction

Theoretical analysis and simulation of complicated processes in catalytic monoliths widely used in industries, are currently active research topics. The aim of this work is modeling and developing robust numerical methods for simulation of physical-chemical processes in a single channel, as a first step to study the complex ones in the whole catalytic monoliths. Earlier investigations can be found in [1–4]. Boundary layer theory is applied for approximation of the flow field. The Navier-Stokes equations are simplified to get a system of large-scale parabolic partial differential equations (PDEs). For simplicity of presentation without loss of generalization, we limit our discussion of reacting flows without surface processes—no catalyst on the surface.

This paper is organized as follows. The mathematical model using the boundary layer equations are stated in Section 2. In Section 3, numerical methods are discussed. Some computational experiments are given in Section 4. The paper concludes in Section 5 with a summary of the obtained results and discussion of further work.

* We would like to thank Dr. Luba Maier, Institute for Chemical Technology and Polymer Chemistry, University of Karlsruhe for supplying the reaction mechanisms and helpful discussions.

2 Mathematical Model

2.1 Modeling the Fluid Dynamical Process

To model flows in a channel, one can employ the boundary layer equations which are simplified from the Navier-Stokes equations. Since our considered channel is an axisymmetrical cylinder, we assume a simplification that the investigated flow in it is also axisymmetrical, which can be described by two spatial coordinates, namely the axial one z and the radial one r . By applying von Mises transformation

$$\psi = \int_0^r \rho u r' dr',$$

where ψ is stream variable and u is axial velocity, there arises the following equation system, which we use for modeling reactive flow in a channel of monoliths [5].

Momentum:

$$\rho u \frac{\partial u}{\partial z} + \frac{\partial p}{\partial z} = \rho u \frac{\partial}{\partial \psi} \left(\rho u \mu r^2 \frac{\partial u}{\partial \psi} \right), \quad (1)$$

$$\frac{\partial p}{\partial \psi} = 0. \quad (2)$$

Energy:

$$\begin{aligned} \rho u c_p \frac{\partial T}{\partial z} = \rho u \frac{\partial}{\partial \psi} \left(\rho u \lambda r^2 \frac{\partial T}{\partial \psi} \right) - \sum_{k=1}^{N_g} \dot{\omega}_k W_k h_k \\ - \rho u r \sum_{k=1}^{N_g} J_{kr} c_{pk} \frac{\partial T}{\partial \psi}. \end{aligned} \quad (3)$$

Species:

$$\rho u \frac{\partial Y_k}{\partial z} = \dot{\omega}_k W_k - \rho u \frac{\partial}{\partial \psi} (r J_{kr}), \quad (k = 1, \dots, N_g). \quad (4)$$

State:

$$p = \frac{\rho R T}{\bar{W}}. \quad (5)$$

Where z and r are the cylindrical coordinates, u and v are the axial and radial components of the velocity vector, p is the pressure, T is the temperature, Y_k is the mass fraction of the k th species, μ is the viscosity, ρ is the mass density, c_p is the heat capacity of the mixture, λ is the thermal conductivity of the mixture, c_{pk} is the specific heat capacity of the k th species, N_g is the total number of gas phase species, K_g is the total number of elementary reactions, J_{kr} is the radial component of mass flux vector of the k th species, $\dot{\omega}_k$ is the rate of production of the k th species by the gas phase reactions, h_k is the specific heat enthalpy of the k th species, W_k is the molecular weight of the k th species, \bar{W} is the mixture mean molecular weight, and R is the universal gas constant.

The diffusion mass flux J_{kr} appearing in (3) and (4) is given by

$$J_{kr} = -D_k^m \frac{W_k}{W} \rho^2 u r \frac{\partial X_k}{\partial \psi} - D_k^T \frac{\rho u r}{T} \frac{\partial T}{\partial \psi}, \quad (6)$$

where D_k^m and D_k^T are diffusion coefficients. The effective diffusion coefficient D_k^m of the k th species into a mixture is approximated by the following mixture-averaged diffusion coefficient formula (see e.g., [1], [6] and [7])

$$D_k^m = \frac{1 - Y_k}{\sum_{i=1, j \neq k}^{N_g} X_k / D_{ik}}, \quad (7)$$

where Y_k and X_k are the mass fraction and the mole fraction of the k th species, and D_{ik} is the binary diffusion coefficients. The binary diffusion coefficients D_{ik} is also estimated based on the logarithm of its value, which is evaluated by the fourth-order polynomial of the logarithm of the temperature

$$\begin{aligned} \ln D_{ik} = & d_{1,ik} + d_{2,ik} \ln T + d_{3,ik} (\ln T)^2 \\ & + d_{4,ik} (\ln T)^3 + d_{5,ik} (\ln T)^4 - \ln p, \end{aligned}$$

where p is the pressure and $d_{l,ik}$ are the coefficients of the fitted polynomial of the binary diffusion coefficients.

As mass fraction, Y_k must satisfy

$$0 \leq Y_k \leq 1 \quad (k = 1, \dots, N_g), \quad \sum_{k=1}^{N_g} Y_k = 1. \quad (8)$$

The relation between ψ and r will be treated later in the differential form

$$0 = \frac{\partial r^2}{\partial \psi} - \frac{2}{\rho u}. \quad (9)$$

2.2 Modeling the Chemical Process

To model the chemical kinetics of moving gas mixture, we use the detailed chemistry describing elementary reactions on molecular level (see, e.g., [8, 9] for more details). The rate of production $\dot{\omega}_k$ of k th species appearing in equations (3) and (4) is given by

$$\dot{\omega}_k = \sum_{i=1}^{K_g} v_{ki} k_{fi} \prod_{j=1}^{N_g} [X_j]^{v'_{ji}} \quad (k = 1, \dots, N_g),$$

where

- K_g : the total number of elementary reactions,
- N_g : the total number of gas phase species,
- v'_{ki} : the stoichiometric coefficient of the k th species in the i th reaction at the *left* side,
- v''_{ki} : the stoichiometric coefficient of the k th species in the i th reaction at the *right* side,
- v_{ki} : $= v''_{ki} - v'_{ki}$,
- $[X_j]$: the concentration of the j th species,
- k_{fi} : the forward rate coefficient of the i th reaction.

The forward rate coefficient k_{fi} is calculated by the Arrhenius expression:

$$k_{fi} = A_i T^{\beta_i} \exp\left(-\frac{E_{ai}}{RT}\right) \quad (i = 1, \dots, K_g),$$

where

- A_i : the pre-exponential factor of the i th reaction
[the units are given in terms of m, mol, and s],
- β_i : temperature exponent of the i th reaction,
- E_{ai} : activation energy of the i th reaction [J],
- R : universal gas constant being 8.314 [J/(mol · K)],
- T : gas temperature [K].

2.3 Initial and Boundary Conditions

Mention that we consider the system only at steady state.

For initial conditions, the value of u , p , T , and Y_k are specified at the inlet of channel.

At $\psi = 0$, which corresponds to the centerline of cylindrical channel, we can deduce the following boundary conditions from the assumed axisymmetry

$$r = 0, \quad \partial u / \partial \psi = 0, \quad \partial p / \partial \psi = 0, \quad \partial T / \partial \psi = 0, \quad \partial Y_k / \partial \psi = 0.$$

At $\psi = \psi_{\max} = \int_0^{r_{\max}} \rho u r' dr' \Big|_{z=0}$, which corresponds to the channel wall, it holds

$$r = r_{\max}, \quad u = 0, \quad T = T_{\text{wall}}, \quad J_{kr} = 0.$$

3 Numerical Methods

3.1 Semi-discretization

We choose the approach of semi-discretization of the PDE system in the direction ψ by the *method of lines* [10]. Here, we have two independent spatial variables z and ψ , but do not have the independent variable “time”. The axial direction z is now treated as the time-like direction.

Using subscript for denoting partial derivatives and the abbreviations

$$\mathcal{E} = \begin{bmatrix} \rho u u_z + p_z \\ 0 \\ \rho u c_p T_z \\ 0 \\ \rho u Y_{1z} \\ \vdots \\ \rho u Y_{N_g z} \end{bmatrix}$$

$$\mathcal{F} = \begin{bmatrix} \rho u (\rho u \mu r^2 u_\psi)_\psi \\ p_\psi \\ \rho u (\rho u \lambda r^2 T_\psi)_\psi - \sum_{k=1}^{N_g} \dot{\omega}_k W_k h_k - \rho u r \sum_{k=1}^{N_g} J_{k,r} c_{pk} T_\psi \\ \frac{\partial r^2}{\partial \psi} - \frac{2}{\rho u} \\ \dot{\omega}_1 W_1 - \rho u (r J_{1,r})_\psi \\ \vdots \\ \dot{\omega}_{N_g} W_{N_g} - \rho u (r J_{N_g,r})_\psi \end{bmatrix}$$

equations (1)- (4) can be summarized to the following system

$$\mathcal{E} = \mathcal{F}, \quad (10)$$

The considered interval of ψ is discretized by ψ_i , $i = 1, \dots, N$. Let us denote the function section corresponding to $\psi = \psi_i$ by the subscript i . For instance,

$$u_i := u_i(z) := u(z, \psi_i).$$

This rule is also applied to partial derivatives, e.g.,

$$u_{\psi_i} := u_{\psi_i}(z) := \left. \frac{\partial u(z, \psi)}{\partial \psi} \right|_{\psi=\psi_i}$$

and other quantities, such as temperature T , pressure p , radial coordinate r , and mass fraction Y_k .

Let $\mathcal{A} = (A_{j,k})$ be the matrix defined by

$$A_{j,k} = \begin{cases} \rho u, & \text{if } j = k = 1 \text{ and } 5 \leq j = k \leq N_g + 4 \\ 1, & \text{if } j = 1, k = 2 \\ \rho u c_p, & \text{if } j = k = 3 \\ 0, & \text{otherwise.} \end{cases}$$

and let

$$\mathcal{Q} = [u, p, T, r, Y_1, Y_2, \dots, Y_{N_g}].$$

Then we have

$$\mathcal{E} = \mathcal{A} \mathcal{Q}_z^T. \quad (11)$$

By our convention,

$$\mathcal{E}_i = \mathcal{E}|_{\psi=\psi_i}, \mathcal{A}_i = \mathcal{A}|_{\psi=\psi_i}, \mathcal{Q}_i = \mathcal{Q}|_{\psi=\psi_i}, \mathcal{Q}_{zi} = \mathcal{Q}_z|_{\psi=\psi_i}.$$

With

$$\begin{aligned} E &= [\mathcal{E}_1^T, \mathcal{E}_2^T, \dots, \mathcal{E}_{N-1}^T]^T, \\ A &= \text{diag}(\mathcal{A}_i), \\ Q &= [\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{N-1}], \\ Q_z &= [\mathcal{Q}_{z1}, \mathcal{Q}_{z2}, \dots, \mathcal{Q}_{zN-1}], \end{aligned}$$

(11) implies

$$E = A Q_z^T,$$

which is the discretization result of the left-hand side of equation (10).

\mathcal{A}_i depends on ρ_i , u_i , and c_{p_i} , which in turn depend only on Q_i . Note that \mathcal{A}_i , $i = 1, \dots, N-1$, are band matrices with upper bandwidth equal to 1 and lower bandwidth equal to 0. Therefore, A inherits this property, too.

We use the forward finite difference to approximate p_ψ :

$$p_{\psi_i} = \frac{p_{i+1} - p_i}{\psi_{i+1} - \psi_i}. \quad (12)$$

Central finite differences are applied to the following partial derivatives with respect to ψ :

$$u_\psi, T_\psi, (\rho u \mu r^2 u_\psi)_\psi, (\rho u \lambda r^2 T_\psi)_\psi, (r J_{kr})_\psi, \text{ and } X_{k\psi}. \quad (13)$$

(Note that $X_{k\psi}$ does appear in J_{kr} as given in (6).)

The fourth component of \mathcal{F} is discretized by the trapezoidal rule:

$$\left[\frac{\partial r^2}{\partial \psi} - \frac{2}{\rho u} \right]_i = \frac{r_i^2 - r_{i-1}^2}{\psi_i - \psi_{i-1}} - \frac{4}{\rho_i u_i + \rho_{i-1} u_{i-1}}. \quad (14)$$

In addition, we have the boundary condition $r_1 = 0$.

Let F_i denote the semi-discretized form of $\mathcal{F}_i = \mathcal{F}|_{\psi=\psi_i}$ by using the approximation scheme described in (12), (13), and (14). Then

$$F = [F_1^T, F_2^T, \dots, F_{N-1}^T]^T$$

is the discretization result of the right-hand side of equation (10). Note that $F = F(Q)$. Due to the central finite difference scheme, F_i depends on the values at three points ψ_{i-1} , ψ_i , and ψ_{i+1} , i.e., $F_i = F_i(Q_{i-1}, Q_i, Q_{i+1})$.

Hence, the PDE system (10) corresponds to

$$A(Q) Q_z^T = F(Q). \quad (15)$$

Initial conditions are

$$u = u_0, p = p_0, T = T_0, Y_k = Y_{k0} \quad (k = 1, \dots, N_g) \text{ at } z = 0. \quad (16)$$

At the channel wall, u , T , p , and r must fulfill

$$0 = \begin{bmatrix} u \\ p \\ T \\ r \\ J_{kr} \end{bmatrix}_{\psi=\psi_N} - \begin{bmatrix} 0 \\ p_{N-1} \\ T_{\text{wall}} \\ r_{\text{max}} \\ 0 \end{bmatrix}. \quad (17)$$

Finally, the equations (15)-(17) together form the Differential Algebraic Equation (DAE) system with the unknowns

$$[Q_1, Q_2, \dots, Q_N].$$

It is worth to say that the partial derivatives of the left- and right-hand side of the DAE with respect to the unknowns and the iteration matrix are of band structure, with total bandwidth $3 \times \dim(Q_i)$. The mentioned block tridiagonal structure arises from choosing suitable indices for Q and F . It is used for efficient computation and storage of derivatives and iteration matrix.

The PDE model equations presented in Section 2 are semi-discretized by using the method of lines leading to a structured system of differential-algebraic equations (DAEs).

3.2 Tailored Solution for the DAE

The DAEs are solved by an implicit method, based on the backward differentiation formulas (BDF). For the practical computation, based on the code DAESOL [11, 12], we develop a new code that allows us to solve this problem efficiently. Features of this code are variable step size and variable order controlled by error estimation, modified Newton method for the solution of the implicit nonlinear problems, an efficient monitoring strategy to control the computation and decomposition of the Jacobian—the *iteration matrix*.

With a suitable arrangement of variables and function models (e.g., natural order scheme), the iteration matrices of the Newton iteration at each step of integration of the DAEs are of block tridiagonal structure. Solution of DAEs by BDF methods requires the partial derivatives of the DAE model functions with respect to the state variables.

For our DAE system, the functions of the DAE are complex involving chemical source terms, diffusion coefficients..., and evaluation of these functions are time-consuming. Therefore, the approximation of the partial derivatives of the DAE with respect to the state variables for forming the Jacobian are very expensive. In general, the Jacobian changes very little during the Newton iteration and even between several integration steps. The efficient monitoring strategy of the computation and decomposition of the Jacobian is developed and implemented. For more details, see [5]. Moreover, the costly part of evaluation of the DAE functions is the evaluation of the diffusion coefficients D_k^m in (7). We see that these coefficients change very little or even nearly the same during the evaluation of the Jacobian. Therefore, instead of re-evaluation of these diffusion coefficients during the computation of the partial derivatives, we keep using the old value of these diffusion coefficients, which are evaluated already during the Newton iteration.

For computation of the (partial) derivatives of the model functions with respect to the state variables, we take the block tridiagonal structure into account. We only need $3 \times (N_g + 4) + 1$ function evaluations for approximation of the Jacobian by the forward finite difference. Alternatively, the partial derivatives can be computed using automatic differentiation technique and exploiting the sparse structure using ADIFOR [13] investigated in [5].

In particular, the following monitoring strategy is implemented:

- (I1) We compute an estimate the convergence ratio δ of the Newton iteration. If $\delta \leq \delta_{\max}$, e.g., $\delta_{\max} = 1/3$, the decomposed iteration matrix is frozen.
- (I2) If the convergence ratio δ is large which indicates a big change of the BDF-coefficients and of the step size, then a new iteration matrix is formed and decomposed using the new BDF-coefficients and the current step size, but the partial derivatives of the DAE functions are not re-computed, keep using the old values.
- (I3) If with the iteration matrix in (I2) the Newton iteration does not converge, this could indicate a big change in the partial derivatives of the DAE functions, then we re-evaluate these partial derivatives and compute and decompose a new iteration matrix. When computing the partial derivatives, the diffusion coefficients D_k^m are not re-evaluated and take the old values.
- (I4) If with the iteration matrix in (I3) the Newton iteration still does not converge, this could indicate that the predicting starting value for the Newton iteration is not sufficiently close to the solution, i.e., the current step size is too big. We reduce the step size.

4 Numerical Results

We investigate a practical application—oxidation of Iso C_8H_{18} using two very large gas phase mechanisms:

- (a) Delphi mechanism: is developed by Delphi Inc., consists of 421 species and 3610 reactions.
- (b) LLNL mechanism: is developed by Lawrence Livermore National Lab, consists of 857 species and 7190 reactions.

In the following, all computations are performed on a AMD Opteron 64, 2 x 1.4 GHz, Linux with Intel Fortran compiler, and computation using double precision. The integration error is controlled with the relative error tolerance $RTOL = 10^{-3}$ and the absolute error tolerance $ATOL = 10^{-9}$. **FD** is the abbreviations for Finite Differences.

Let us define the **Speedup** to be the ratio between the CPU time for running the simulation, of which the diffusion coefficients are computed during the calculation of the partial derivatives, called *standard* mode, and the CPU time for running the simulation, of which the diffusion coefficients were taken the old values during the calculation of the partial derivatives, called *frozen* mode. The comparing the performance of different methods of calculating of the partial derivatives for solution of the DAE is reported in [5].

Tables 1 and 2 show CPU times and Speedup for simulation with the two modes of derivatives computation for the two mechanisms (a) and (b) as above. The **Nodes** stands for the number of grid points in the stream direction. The frozen mode is few times faster than the standard mode. The speedup is from 4 to 10. There is a trend in speedup gain with respect to the size of mechanism (the number of species).

Computational statistics are shown in Tables 1 and 2, where **#LA** is the total number of times for solving linear equation, is also the total number of Newton iterations for all integration steps. **#PD Eval.** is total number partial derivative evaluations, also the number of Jacobian evaluations. **# Jac. Decomp.** is the total number of Jacobian

decompositions. This results show that the cost of evaluation of the partial derivatives at the step (I3) in Section 3.2 are dominated and significantly decreased when the diffusion coefficients are kept frozen, and the total computational times have been significant reduced. The convergence of the modified Newton iterations nearly are not affected by the frozen of the diffusion coefficients.

Table 1. Timings with two modes of derivatives computation of Delphi’s mechanism

Nodes	12	14	16
Standard Blk.Tri. FD	4h3’	4h54’	5h51’
Frozen Blk.Tri. FD	58’	1h13’	1h36’
Speedup	4.18	4.02	3.65

Table 2. Timings with two modes of derivatives computation of LLNL’s mechanism

Nodes	7	8	9
Blk.Tri. FD	10h31’	12h36’	15h25’
Frozen Blk.Tri. FD	59’	1h 22’	2h3’
Speedup	10.69	9.21	7.52

Table 3. Computational statistics with two modes of derivatives computation of Delphi’s mechanism

Node	Mode	#LA	#Step	#PD Eval.	# Jac. Decomp.
12	Blk.Tri. FD	332	110	13	52
	Frozen Blk.Tri. FD	332	110	13	52
14	Blk.Tri. FD	321	108	13	48
	Frozen Blk.Tri. FD	321	108	13	48
16	Blk.Tri. FD	327	109	13	51
	Frozen Blk.Tri. FD	327	109	13	51

5 Conclusion

We have shown efficient numerical methods for simulation of chemically reacting flows in a single channel of catalytic monoliths. By exploiting the structure of semi-discretization of the PDEs model, i.e., of block tridiagonal structure and frozen the diffusion coefficients, we can obtain the speedup of simulation code up to 10. The new simulation software is applied to a practical application, oxidation of Iso C_8H_{18} . Further developments of the methods to optimization of chemically reacting flows in catalytic monoliths were also considered, and will be published elsewhere.

Table 4. Computational statistics with two modes of derivatives computation of LLNL's mechanism

Node	Mode	#LA	#Step	#PD eval.	# Jac. Decomp.
7	Blk.Tri. FD	467	171	8	40
	Frozen Blk.Tri. FD	457	168	8	41
8	Blk.Tri. FD	456	171	8	42
	Frozen Blk.Tri. FD	461	172	8	41
9	Blk.Tri. FD	456	170	8	41
	Frozen Blk.Tri. FD	456	170	8	41

References

1. Kee, R.J., Miller, J.A.: A computational model for chemically reacting flow in boundary layers, shear layers, and ducts. Technical Report SAND81-8241, Sandia National Laboratories, Albuquerque, NM (1981)
2. Coltrin, M.E., Kee, R.J., Miller, J.A.: A mathematical model of the coupled fluid mechanics and chemical kinetics in a chemical vapor deposition reactor. *Journal of The Electrochemical Society* **131**(2) (1984) 425–434
3. Coltrin, M.E., Kee, R.J., Miller, J.A.: A mathematical model of silicon chemical vapor deposition. *Journal of The Electrochemical Society* **133**(6) (1986) 1206–1213
4. Coltrin, M.E., Moffat, H.K., Kee, R.J., Rupley, F.M.: CRESLAF (version 4.0): A Fortran program for modelling laminar, chemically reacting, boundary-layer flow in the cylindrical or planar channels. Technical Report SAND93-0478, Sandia National Laboratories (Apr 1993)
5. Minh, H.D.: Numerical Methods for Simulation and Optimization of Chemically Reacting Flows in Catalytic Monoliths. PhD thesis, Faculty of Mathematics and Computer Science, University of Heidelberg (December 2005)
6. Warnatz, J.: Influence of transport models and boundary conditions on flame structure. In Peters, N., Warnatz, J., eds.: *Numerical Methods in Flame Propagation*. Friedr. Vieweg and Sohn, Wiesbaden (1982)
7. Kee, R.J., Warnatz, J., Miller, J.A.: A Fortran computer code package for the evaluation of gas phase viscosities, heat conductivities, and diffusion coefficients. Technical Report SAND83-8209, Sandia National Laboratories (1983)
8. Warnatz, J., Dibble, R., Maas, U.: *Combustion, Physical and Chemical Fundamentals, Modeling and Simulation, Experiments, Pollutant Formation*. Springer-Verlag, New York (1996)
9. O.Deutschmann, Tischer, S., Kleditzsch, S., V.M.Janardhanan, Correa, C., Chatterjee, D., Mladenov, N., Minh, H.D.: DETCHEM - User's manual. www.detchem.com. (2007)
10. Schiesser, W.E.: *The Numerical Method of Lines, Integration of Partial Differential Equations*. Academic Press, San Diego, CA (1991)
11. Bauer, I., Bock, H.G., Schlöder, J.P.: DAESOL – a BDF-code for the numerical solution of differential algebraic equations. Technical report, SFB 359, IWR, University of Heidelberg (1999)
12. Bauer, I., Finocchi, F., Duschl, W., Gail, H., Schlöder, J.: Simulation of chemical reactions and dust destruction in protoplanetary accretion disks. *Astronomy & Astrophys.* **317** (1997) 273–289
13. Bischof, C., Carle, A., Hovland, P., Khademi, P., Mauer, A.: ADIFOR 2.0 User's Guide. (1995)