The R Package diseq: Estimation Methods for Markets in Equilibrium and Disequilibrium

This version: April 29, 2021

Pantelis Karapanagiotis[1]

Abstract

Market models constitute a major cornerstone of empirical research in industrial organization and macroeconomics. Previous literature in these fields has proposed a variety of estimation methods both for markets in equilibrium, which typically entail a market-clearing condition, and in disequilibrium, in which the primary identification condition comes from the short-side rule. Although methodologically attractive, the estimation methods of such models, in particular of the disequilibrium models, is computationally demanding and software providing simple, out-of-the-box methods for estimating them is scarce. Econometricians, therefore, mostly rely on their own implementations for estimating these models. This article presents the `R` package diseq, which provides functionality to simplify the estimation of models for markets in equilibrium and disequilibrium using full information maximum likelihood methods. The basic functionality of the package is presented based on the data and the classic analysis originally performed by Fair & Jaffee (1972). The article also gives an overview of the design of the package, presents the post-estimation analysis capabilities that accompany it, and provides statistical evidence of the computational performance of its functionality gathered via large-scale benchmarking simulations. Diseq is free software that is distributed under the MIT license as part of the `R` software project. It comprises a set of estimation tools, which are to a large extend not available from either alternative `R` packages or other statistical software projects.

JEL-Classification: C13, C34, C51, C87, D50, L10,

Keywords: market clearing, shortages, disequilibrium, marginal effects, market aggregates, maximum likelihood

[1]Goethe University Frankfurt, Theodor-W.-Adorno Platz 4, 60323 Frankfurt, Germany; Tel.: +49 69 798 30081; Pantelis.Karapanagiotis@hof.uni-frankfurt.de

1. Introduction

Demand and supply estimations are among the most common objectives of the econometrics involved in policy-making and academic analysis. These two market forces are typically interdependent through prices, which affect both sides of the market. Thus, it is well understood in the economic literature that estimating separately demand or supply has endogeneity issues. Therefore, the majority of estimation methods for demand and supply account that these market forces are simultaneously determined and model them after systems of potentially dynamic, stochastic, simultaneous equations.

Estimation methods for systems of market equations adopt a variety of structural assumptions about the market. The most common assumption is that of market-clearing, which postulates that prices are infinitely responsive to demand and supply changes and swiftly adjust so that demand and supply meet each other. Economists have employed alternative structural assumptions in studying situations for which the market-clearing assumption constitutes a poor approximation. Unemployment in labor markets, financial constraints in credit markets, and shortages in agricultural products are more commonly studied by replacing the market-clearing assumption with the short-side rule. The short-side rule presumes that price adjustment is not perfect, and temporary market imbalances can lead to disagreements between demanded and supplied quantities.

Methodologies for estimating markets using either market-clearing or the short-side rule as the primary structural identification condition are known for years. Still, because the short-side rule introduces non-linearities to the system of market equations, the computational difficulties relating to methodologies for markets in disequilibrium posed a bottleneck in the application of such methods, also hindering in this way the development of statistical software implementing them. This article introduces the statistical `R` package diseq, which closes this long-lasting gap by providing a common framework for estimation methods and analysis tools for markets in equilibrium and disequilibrium.

The common framework allows the estimation of five market models; a model with market-clearing and four models for markets in disequilibrium. Except for the equilibrium model, which can also be estimated using two-stage least squares, the models of diseq are estimated by maximizing the full information likelihood. The common estimation interface allows interchanging among the available optimization methods in `optimr`. Access to native optimization procedures from `GSL` are also provided for the equilibrium model. More importantly, diseq uses analytic expressions for calculating the gradients of all models' likelihoods by default, which greatly enhances the computational performance of maximum likelihood optimization. The article presents statistical evidence in favor of the overperformance gathered from large-scale benchmarking estimations using simulated data in the high-performance cluster of Goethe University's Center for Scientific Computing (CSC). Specifically, benchmarking measurements for estimating all models using `BFGS` with analytically calculated gradients, `BFGS` with numerically approximated gradients, and Nelder-Mead are presented. In addition, diseq calculates standard errors using analytic Hessian expressions by default for two of the disequilibrium models. The package also

provides options for estimating clustered standard errors or adjusting for heteroscedasticity. The option of using analytic gradient and Hessian expressions in the estimation calls is not available in any alternative statistical software package.

Besides providing a unified and computationally efficient way to estimate various market models, diseq provides a minimalistic set of post-estimation analysis methods. These simplify the calculation of (dis-)aggregated predicted demanded and supplied values, various measures of shortages, and marginal effects.

This article gives an overview of data analysis with the `R` package diseq. Section 2 demonstrates the typical usage of diseq via the classic analysis of Fair & Jaffee (1972). Section 3 offers a top-down overview of the design of the package. Section 4 describes the functions provided by the diseq package for estimating and analyzing markets in equilibrium and disequilibrium. Section 4 discusses alternative tools for estimating markets in disequilibrium and provides some computational benchmarks. The last section concludes.

2. An empirical example

2.1. The houses dataset

The `houses` dataset contains the monthly macroeconomic time series for the US credit market of housing starts from July 1958 to December 1969. The market for this period was initially studied by Fair (1971). Subsequent work on estimation of markets with disequilibrium methods, by Fair & Jaffee (1972), Maddala & Nelson (1974), and Hwang (1980), also uses the housing market of this period for illustrating the introduced methodologies.

Table 1 presents the variables of the `houses` dataset and provides a short description for each one of them. The observations of HS were collected from the *Economic Report of the President* (1947). The series of RM were obtained by Fair (1971, table A.3). The observations of W were manually collected, and the data for $DSLA$, $DMSB$, and $DHLB$ were collected from the *Federal Reserve Bulletin* (1914).

Table 1: Variables in the houses dataset.

$DATE$	The date of the record.
HS	Private non-farm housing starts in thousands of units (not seasonally adjusted).
RM	FHA Mortgage Rate series on new homes in units of 100 (beginning-of-month Data).
$DSLA$	Savings capital (deposits) of savings and loan associations in millions of dollars.
$DMSB$	Deposits of mutual savings banks in millions of dollars.
$DHLB$	Advances of the federal home loan bank to savings and loan associations in million of dollars.
W	Number of working days in month.

The specification of the demand and supply equations follows Hwang (1980) and Maddala & Nelson (1974).

The demand equation is given by

$$D_t = \beta_0^d RM_t + \beta_1^d + \beta_2^d t + \beta_3^d W_t + \beta_4^d CSHS_t + \beta_5^d RM_{t-1} + \beta_6^d RM_{t-2} + \sum_{i=2}^{12} \beta_{5+i}^d MONTH_{i,t} + u_t^d,$$

where $CSHS$ is the cumulative sum of past housing starts and $MONTH_i$ are monthly indicator variables. The variable $CSHS$ is used as a proxy to the stock of houses, and the monthly indicators are used to capture seasonal demand effects. The supply equation is specified as

$$S_t = \beta_0^s RM_t + \beta_1^s + \beta_2^s t + \beta_3^s W_t + \beta_4^s RM_{t-1} + \beta_4^s MA_6(DSF_t) + \beta_6^s MA_3(DHF_t) + \sum_{i=2}^{12} \beta_{5+i}^s MONTH_{i,t} + u_t^s,$$

where $MA_6(DSF)$ is the moving average of order 6 of the flow of deposits in savings associations and loan associations and mutual savings banks, and $MA_3(DHF)$ is the moving average of order 3 of the flow of advances of the federal home loan bank to savings and loan associations. The stochastic terms $u_{d,t}$ and $u_{s,t}$ are jointly, normally distributed. The moving averages and lagged variables of the analysis are constructed from the variables of the `houses` dataset. The `fair_houses` function of diseq automates the construction (see listing 1).

Listing 1: Data preparation.

```
1 house_data <- fair_houses()
```

2.2. Model initialization

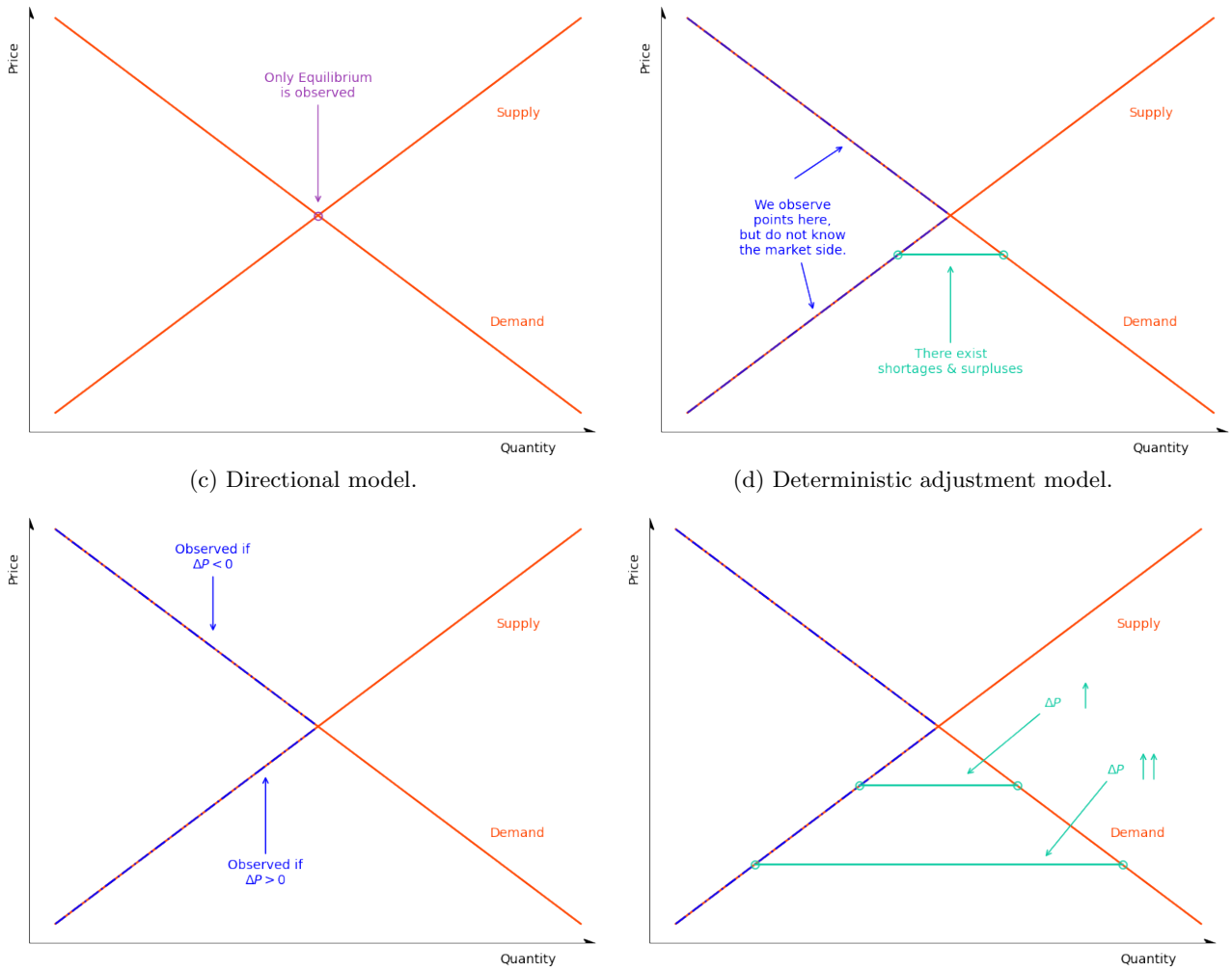
The demand and supply equations are estimated using four market structures. Each market structure entails a different assumption about the price dynamics of the market. Figure 1 illustrates the differences between the four models. The equilibrium model (fig. 1a) assumes that market prices seamlessly adjust to demand and supply changes, and the econometrician observes only equilibrium points in the data, i.e., $HS_t = D_t = S_t$ for each t in the sample. The basic disequilibrium model (fig. 1b) assumes that the observed quantity in the data is determined by the short side rule, i.e., $HS_t = \min\{D_t, S_t\}$. The directional model (fig. 1c) add a separation rule to the identification assumption of the basic model. The sample is separated in demand and supply observations based on the observed price changes. If $MR_t - MR_{t-1} \geq 0$, then the observation at time t is classified as belonging to the supply side, otherwise it is classified as a demand side observation. Finally, the deterministic adjustment model (fig. 1b) additionally assumes that price changes are analogous to the shortages and surpluses of the market, i.e., $\gamma\,(MR_t - MR_{t-1}) = D_t - S_t$ for some positive parameter γ that is to be estimated.

The initialization arguments of the constructors of the four models mostly coincide. Each model initialization requires specifying the model class, the used dataset, the identifiers of the dataset, the quantity and price variables, and the demand and supply specifications. If the initialized model involves price dynamics, as is the case for the directional and deterministic adjustment model, the construction operation also requires specifying

Figure 1: Market specifications.

(a) Equilibrium model.

(b) Basic model.



(c) Directional model.

(d) Deterministic adjustment model.

the dataset's time column. Additionally, one can choose whether the initialized model should allow for temporal correlation between the shocks of the demand and supply equations and the verbosity level with which the operations of the constructed model should emit messages to the user. The initialization options across the four models used in this example are kept similar to allow for comparability. The common input variables are defined in Listing 2.

Listing 2: Model specification.

```
key_columns <- c("ID", "TREND")
quantity_column <- "HS"
price_column <- "RM"
time_column <- "TREND"
```

```
demand_specification <- "TREND + W + CSHS + L1RM + L2RM + MONTH"
supply_specification <- "TREND + W + L1RM + MA6DSF + MA3DHF + MONTH"
correlated_shocks <- FALSE
verbose <- 3
```

Initialization requires specifying two key columns in order to handle panel data. For panel data, one key column should specify the entity identifier, and the other key column should specify the time identifier. As in the case of this example's data, time series data can be handled by passing a constant entity identification value for all observations of the used sample. The indicator variables of `factor` type columns included in the market equations, such as the $MONTH$ variable of this example, are automatically created by the constructors, and the variable that corresponds to the first level of the factor is dropped.

Listing 3 contains the models' initialization code. The `equilibrium_model` and `diseq_basic` models are atemporal, and thus their constructors do not require specifying a time column. The `diseq_deterministic_adjustment` and `diseq_directional` models entail price dynamics, and the `time_column` variable is passed as the third argument to their constructors. Expect for the case of the `diseq_directional`, the `price_column` is added to the `demand_specification` and `demand_specification`. The directional model uses the `price_column` for separating the sample without adding any additional degree of freedom in its estimation and, thus, the price variables cannot be part of both the demand and supply sides (Maddala & Nelson, 1974, p. 1021).

Listing 3: Model initialization.

```
equilibrium_mdl <- new(
  "equilibrium_model",
  key_columns, quantity_column, price_column,
  paste(price_column, demand_specification, sep = "+"),
  paste(price_column, supply_specification, sep = "+"),
  house_data,
  correlated_shocks = correlated_shocks, verbose = verbose)
basic_mdl <- new(
  "diseq_basic",
  key_columns, quantity_column, price_column,
  paste(price_column, demand_specification, sep = "+"),
  paste(price_column, supply_specification, sep = "+"),
  house_data,
  correlated_shocks = correlated_shocks, verbose = verbose)
directional_mdl <- new(
  "diseq_directional",
  key_columns, time_column, quantity_column, price_column,
  paste(price_column, demand_specification, sep = "+"),
```

```
  supply_specification,
  house_data,
  correlated_shocks = correlated_shocks, verbose = verbose)
deterministic_adjustment_mdl <- new(
  "diseq_deterministic_adjustment",
  key_columns, time_column, quantity_column, price_column,
  paste(price_column, demand_specification, sep = "+"),
  paste(price_column, supply_specification, sep = "+"),
  house_data,
  correlated_shocks = correlated_shocks, verbose = verbose)
```

2.3. Estimation

The models of this example are estimated by maximizing their likelihoods (see Maddala & Nelson (1974) and Maddala (1986)). The diseq packages relies mainly on `R` package bbmle for maximizing the likelihoods[2]. By default, likelihoods are maximized using the Broyden-Fletcher-Goldfarb-Shanno (hereafter `BFGS`) algorithm with gradients that are calculated by analytic expressions, but the user can override this behavior. Other options regarding the calculation of the Hessians and standard errors are documented in section 4.

Listing 4 contains the commands for estimating the four models of the example. The two additional keyword arguments given the the estimation call of this example (`optimization_control` and `method`) are passed to the `bbmle::mle2` function. If starting values are not provided, the `estimate` function initializes the optimization routine using as starting values the coefficient estimates obtained by linear regressions of the demand and supply equations of the model. The default estimation method is `BFGS` with analytically calculated gradients. For the basic and directional models, for which Hessian expressions are available, standard errors are calculated using these expressions by default.

Listing 4: Model estimation.

```
optimization_control <- list(maxit = 50000)
equilibrium_est <- estimate(equilibrium_mdl, control = optimization_control)
basic_est <- estimate(basic_mdl, control = optimization_control,
                      start = equilibrium_est@coef)
directional_est <- estimate(directional_mdl, method = "Nelder-Mead",
                            control = optimization_control)
deterministic_adjustment_est <- estimate(deterministic_adjustment_mdl,
                                         control = optimization_control)
```

[2]Additional methods and tools are available for the equilibrium model. See section 3 for more information.

Table 2 presents the estimated coefficients for the four used models. The coefficients of the monthly indicators are omitted for brevity. Parentheses contain the p-values for the estimated coefficients. The price elasticities of demand (D_RM) and supply (S_RM) of the equilibrium, basic, and deterministic adjustment models have the economically expected signs. In contrast, the demand elasticity (D_RM) of the directional model has an opposite sign than the expected one. The deterministic adjustment model suggests that price changes are responsive to shortages and surpluses as the estimated response parameter (RM_DIFF) is greater than one and has a p-value equal to one. The demand and supply equations of the directional model closely resemble the corresponding equations in the analysis of Fair (1971) and Fair & Jaffee (1972), and its estimated coefficients have identical signs and similar scales to the corresponding estimates of those articles.

Table 2: Estimation results.

Coefficients	Equilibrium	Basic	Directional	Deterministic adjustment
D_RM	−5.8707 (0.00)	−8.1205 (0.00)	0.9294 (0.00)	−4.5422 (0.00)
D_CONST	−3.5384 (0.00)	−3.3352 (0.00)	104.2485 (0.00)	13.6044 (0.00)
D_TREND	−2.2583 (0.00)	−34.7577 (0.00)	3.4024 (0.00)	−2.6245 (0.00)
D_W	3.2213 (0.01)	−32.2648 (0.00)	5.4002 (0.00)	2.5180 (0.04)
D_CSHS	0.0211 (0.00)	0.2031 (0.00)	−0.0259 (0.00)	0.0244 (0.00)
D_L1RM	7.8750 (0.00)	14.7104 (0.00)	0.3572 (0.41)	6.2722 (0.00)
D_L2RM	−1.9786 (0.00)	−3.1749 (0.33)	−1.5413 (0.00)	−1.6969 (0.00)
S_RM	0.9617 (0.00)	0.4015 (0.06)	NA (NA)	0.6201 (0.00)
S_CONST	−57.8048 (0.00)	−77.7746 (0.00)	−39.5322 (0.00)	−63.6607 (0.00)
S_TREND	−0.1787 (0.00)	−0.1416 (0.00)	−0.0714 (0.00)	−0.1706 (0.00)
S_W	2.9853 (0.00)	3.2784 (0.00)	3.7903 (0.00)	2.9240 (0.00)
S_L1RM	−0.9008 (0.00)	−0.3259 (0.13)	0.0223 (0.02)	−0.5449 (0.01)
S_MA6DSF	0.0510 (0.00)	0.0522 (0.00)	0.0388 (0.00)	0.0494 (0.00)
S_MA3DHF	0.0408 (0.00)	0.0393 (0.00)	0.0258 (0.00)	0.0343 (0.00)
RM_DIFF	NA (NA)	NA (NA)	NA (NA)	1.6103 (0.01)
$D\_VARIANCE$	805.1174 (0.00)	786.2419 (0.00)	44.1599 (0.00)	810.0375 (0.00)
$S\_VARIANCE$	116.1149 (0.00)	99.5034 (0.00)	25.2621 (0.00)	110.5647 (0.00)

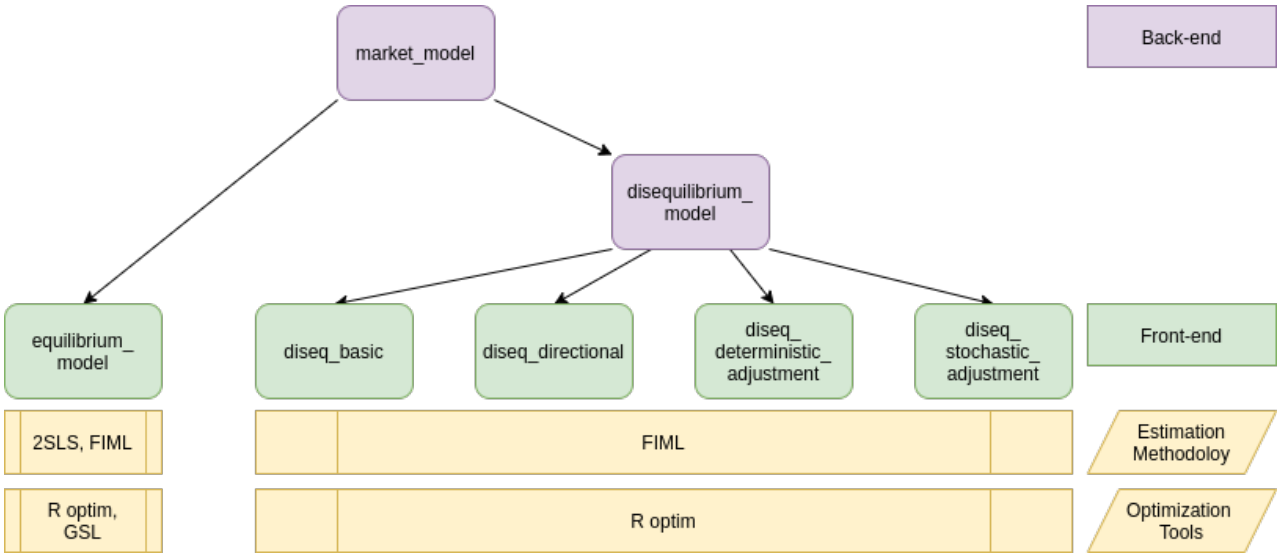
3. Design and statistical background

The package currently supports the estimation of five market models. One of these five models assumes that the studied market preternaturally clears. The remaining four models assume that the studied market switches

between states of excess demand and excess supply. In the disequilibrium models, the market-clearing state has zero probability and can be included in either the excess demand or excess supply state without losing generality. Diseq provides a unified estimation framework for the implemented five models, irrespective of their structural assumptions about the studied market.

The package organizes these classes in a simple object oriented hierarchy that is depicted in fig. 2. Five front-end classes are exposed to the user; these are the (i) `equilibrium_model`, (ii) `diseq_basic`, (iii) `diseq_directional`, (iv) `diseq_deterministic_adjustment`, and (v) `diseq_stochastic_adjustment` classes. The two back-end classes, namely (i) `market_model` and (ii) `disequilibrium_model`, act as interfaces of functionality that is common in market models and disequilibrium market models.

Figure 2: Design overview.



The equilibrium model is represented by a linear system of stochastic equations

$$D_{n,t} = \alpha^d P_{n,t} + \beta_0^d + \sum_{j=1}^{k_d} \beta_j^d X_{n,t}^{j,d} + \sum_{j=1}^{k} \eta_j^d X_{n,t}^{j} + u_{n,t}^d \tag{1}$$

$$S_{n,t} = \alpha^s P_{n,t} + \beta_0^s + \sum_{j=1}^{k_s} \beta_j^s X_{n,t}^{j,s} + \sum_{j=1}^{k} \eta_j^s X_{n,t}^{j} + u_{n,t}^s \tag{2}$$

$$Q_{n,t} = D_{n,t} = S_{n,t}, \tag{3}$$

where D is the demanded quantity, S the supplied quantity, Q the traded quantity, P the price, $X^{j,d}$ and $X^{j,s}$ are equation specific controlled variables, X^j are common controlled variables, and u^d and u^s are jointly, normally distributed shocks. Equation (3) is the market clearing condition that postulates that demanded and supplied quantities are equal.

The equilibrium model can be estimated using either two-stage least squares or full information maximum

likelihood (see Zellner & Theil (1962) and Maddala & Nelson (1974)). The two methods are asymptotically equivalent (Balestra & Varadharajan-Krishnakumar, 1987). Substituting eq. (3) into eqs. (1) and (2) gives a system of two stochastic equations on traded quantities and prices. Suppose that f denotes the joint distribution of u^d and u^s, and let $\theta = \left(\alpha^d, \theta_d, \alpha^s, \theta_s\right)$ and $Y = (Y_d, Y_s)$, where, for $m = d, s$,

$$\theta_m = \left(\beta_0^m, \left\{\beta_j^m\right\}_j, \left\{\eta_j^m\right\}_j\right)' \quad \text{and} \quad Y_m = \left(1, \left\{X_j^{j,m}\right\}_j, \left\{X^j\right\}_j\right)'.$$

Then, the likelihood of the equilibrium model is obtained by

$$\mathrm{L}\left(\theta; q, p, Y\right) = \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q, P\right)}\left(q, p \mid Y; \theta\right) = f\left(q - \alpha^d p - \theta_d' Y_d,\ q - \alpha^s p - \theta_s' Y_s\right).$$

Diseq implements both maximum likelihood and least squares estimation methodologies. The least square estimation uses the systemfit package, while the maximum likelihood methodology uses `optim` via the bbmle, or `Gnu Scientific Library` (henceforth `GSL`) optimization routines.

When building the package from source, If the `C++20` version of `execution.h` is located in the target machine during installation, the gradient calculations are performed in parallel using the `std::execution::par_unseq` execution policy when the `GSL` optimization routines are employed in the estimation. The usage of native optimization routines does not necessarily result in faster execution times because there is an overhead stemming from the communication between `R` and `GSL`. For small datasets, machines without `C++20` support, or machines with few available processors, the communication cost is typically greater than the benefits of using native routines, which results in slower execution times. Still, estimating the equilibrium model via `GSL` routines allows the user to further customize the optimization call by choosing the step size and the gradient tolerance of the `BFGS` algorithm. The two-stage least square is the least computationally intensive estimation methodology as it merely involves linear algebra operations (see Henningsen & Hamann (2007) for the statistical background).

For well-behaved samples, all estimation methods and tools result in similar estimates. Table 3 exemplifies this by comparing the estimates obtained by estimating the equilibrium model using simulated data with 20000 observations. The simulated equilibrium model has, besides prices and a constant, two demand covariates (X_1^d and X_2^d), one supply covariate (X_1^s), two common covariates (X_1 and X_2), and allows for temporal correlation between demand and supply shocks (ρ). The code that generates the sample data and performs the estimations is given in listing 5.

The disequilibrium models keep eqs. (1) and (2) and replace the marketing-clearing condition (eq. (3)) with the short side rule, i.e.,

$$Q_{nt} = \min\left\{D_{n,t}, S_{n,t}\right\}. \tag{4}$$

This modification makes the systems of all the disequilibrium models non linear. The probability that the traded quantity equals the demanded quantity is calculated by

$$\pi_D = \mathbb{P}\left(S > D \mid p, Y; \theta\right) = \mathbb{P}\left(u^s - u^d > \alpha^d p + \theta_d' Y_d - \alpha^s p - \theta_s' Y_s \mid p, Y; \theta\right),$$

Table 3: Comparison of equilibrium estimation methods and tools.

Coefficients	sim	fiml_optim	fiml_gsl	2sls
D_P	−0.70	−0.57 (0.13)	−0.57 (0.13)	−0.57 (0.13)
D_CONST	28.90	27.45 (1.45)	27.40 (1.50)	27.42 (1.48)
D_Xd1	0.30	0.26 (0.04)	0.26 (0.04)	0.26 (0.04)
D_Xd2	−0.20	−0.24 (0.04)	−0.24 (0.04)	−0.24 (0.04)
D_X1	−0.03	0.02 (0.05)	0.02 (0.05)	0.02 (0.05)
D_X2	−0.01	−0.07 (0.06)	−0.07 (0.06)	−0.07 (0.06)
S_P	0.60	0.56 (0.04)	0.56 (0.04)	0.56 (0.04)
S_CONST	10.20	10.53 (0.33)	10.59 (0.39)	10.50 (0.30)
S_Xs1	0.30	0.36 (0.06)	0.35 (0.05)	0.36 (0.06)
S_X1	0.50	0.50 (0.00)	0.50 (0.00)	0.50 (0.00)
S_X2	0.02	0.02 (0.00)	0.02 (0.00)	0.02 (0.00)
$D\_VARIANCE$	4.00	2.96 (1.04)	2.94 (1.06)	2.95 (1.05)
$S\_VARIANCE$	9.00	8.34 (0.66)	8.28 (0.72)	8.38 (0.62)
RHO	−0.30	−0.12 (0.18)	−0.11 (0.19)	−0.12 (0.18)

where the random variables $u^s - u^d$ is normally distributed as difference of normally distributed random variables. The probability that the traded quantity equals the supplied quantity, denoted by π_S, is defined analogously.

The basic model is defined by eqs. (1), (2) and (4), and its likelihood is given by

$$\begin{aligned} \mathrm{L}\left(\theta; q, p, Y\right) &= \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid S > D, p, Y; \theta\right)\pi_D + \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid D \geq S, p, Y; \theta\right)\pi_S \\ &= \int_q^\infty f\left(q - \alpha^d p - \theta_d' Y_d,\ S\right)\mathrm{d}S + \int_q^\infty f\left(D,\ q - \alpha^s p - \theta_s' Y_s\right)\mathrm{d}D. \end{aligned}$$

In addition to eqs. (1), (2) and (4), the directional model separates the sample based on the rule

$$\Delta P \geq 0 \implies D \geq S. \tag{5}$$

Its likelihood is

$$\mathrm{L}\left(\theta; q, p, Y\right) = \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid S > D, p, Y; \theta\right)\pi_D\right)^{1-\mathbb{I}_{\Delta P \geq 0}} \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}Q}\left(q \mid D \geq S, p, Y; \theta\right)\pi_S\right)^{\mathbb{I}_{\Delta P \geq 0}},$$

where $\mathbb{I}_{\Delta P \geq 0}$ is an indicator function taking the value one if eq. (5) is satisfied. The deterministic adjustment model is defined by eqs. (1), (2) and (4), and the deterministic price dynamics

$$\Delta P_{n,t} = \frac{1}{\gamma}\left(D_{n,t} - S_{n,t}\right). \tag{6}$$

Listing 5: Equilibrium estimation methods and tools

```
seed <- 25
parameters <- list(
  nobs = 4000, tobs = 5,
  alpha_d = -0.7, beta_d0 = 28.9, beta_d = c(0.3, -0.2), eta_d = c(-0.03, -0.01),
  alpha_s = 0.6, beta_s0 = 10.2, beta_s = c(0.3), eta_s = c(0.5, 0.02),
  sigma_d = 2.0, sigma_s = 3.0, rho_ds = -0.3)
equilibrium_mdl <- simulate_model("equilibrium_model", parameters, seed, verbose)
fiml_optim_est <- estimate(equilibrium_mdl, control = optimization_control)
fiml_gsl_est <- maximize_log_likelihood(
  equilibrium_mdl, step = .01, objective_tolerance = .01, gradient_tolerance = .01)
ls_est <- estimate(equilibrium_mdl, method = "2SLS")
```

Equation (6) also serves as a separation rule. Given the classification of an observation based on this rule, one of the variables D an S can be eliminated from the system and the remaining variable equals to the traded quantity. This, abusing slightly the notation so that the vector θ also contains γ, results to the likelihood

$$\begin{aligned}
&\mathrm{L} := \mathrm{L}\left(\theta; q, p, p_{-1}, Y\right) \\
&= \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q,P\right)}\left(q,p \mid S > D, p_{-1}, Y; \theta\right)\pi_D\right)^{1-\mathbb{I}_{\Delta P\geq 0}} \left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q,P\right)}\left(q,p \mid D \geq S, p_{-1}, Y; \theta\right)\pi_S\right)^{\mathbb{I}_{\Delta P\geq 0}} \\
&= f\left(q-\alpha^d p-\theta_d' Y_d,\ q-\gamma\Delta p-\alpha^s p-\theta_s' Y_s\right)^{1-\mathbb{I}_{\Delta P\geq 0}} f\left(q-\gamma\Delta p-\alpha^d p-\theta_d' Y_d,\ q-\alpha^s p-\theta_s' Y_s\right)^{\mathbb{I}_{\Delta P\geq 0}}.
\end{aligned}$$

Lastly, the system of the stochastic adjustment model is determined by eqs. (1), (2) and (4), and the stochastic price dynamics

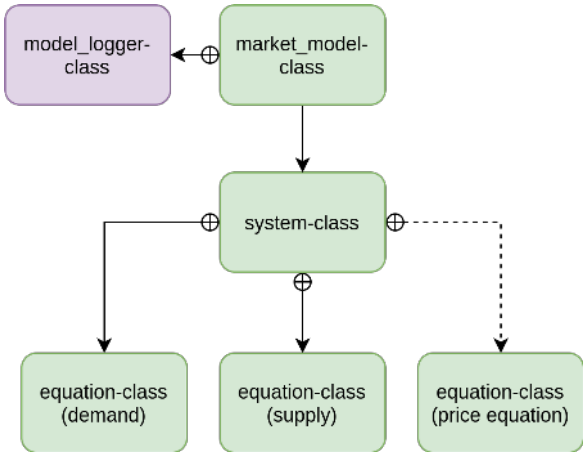
$$\Delta P_{n,t} = \frac{1}{\gamma}\left(D_{n,t} - S_{n,t}\right) + \beta_0^p + \sum_{j=1}^{k_p} \beta_j^p X_{n,t}^{j,p} + u_{n,t}^p, \tag{7}$$

where $X^{j,p}$ are the controlled variables and u^p the disturbance term of the price dynamics. For this model, let $\theta_p = \left(\beta_j^p\right)_j'$, $Y_p = \left(X^{j,p}\right)_j'$. Suppose also that θ contains the parameters γ and θ_p, and that Y contains Y_p. Moreover, let f denote the joint density of the shocks u^d, u^s, and u^p. Since eq. (7) is stochastic, it cannot be used to separate the sample. The likelihood of this model is given by

$$\begin{aligned}
\mathrm{L}\left(\theta; q, p, p_{-1}, Y\right) &= \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q,P\right)}\left(q,p \mid S > D, p_{-1}, Y; \theta\right)\pi_D + \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\left(Q,P\right)}\left(q,p \mid D \geq S, p_{-1}, Y; \theta\right)\pi_S \\
&= \int_q^\infty f\left(q-\alpha^d p-\theta_d' Y_d,\ S,\ \Delta p - \frac{1}{\gamma}\left(q-S\right) - \theta_p' Y_p\right)\mathrm{d}S + \\
&\quad \int_q^\infty f\left(D,\ q-\alpha^s p-\theta_s' Y_s,\ \Delta p - \frac{1}{\gamma}\left(D-q\right) - \theta_p' Y_p\right)\mathrm{d}D.
\end{aligned}$$

Diseq implements maximum likelihood estimation routines for the four disequilibrium models. The likelihoods are optimized using the bbmle package, which relies on `optim`. All five models of diseq have similar implementation components. These are depicted in fig. 3. Each market model contains a logging object that handles the exposed methods' output depending on the level of verbosity that is chosen during initialization. More importantly, it contains an object of its corresponding system class. The system class contains data and methods that describe the system of stochastic equations that describes the model. In turn, the system class contains an equation object for each stochastic equation of the system. Therefore, all models contain a demand and supply equation, and `diseq_stochastic_adjustment` additionally contains a price equation class. Most of the functionality of diseq is exposed via methods at the market class level. The system and equation classes are primarily used to store and organize intermediate and final analysis data.

Figure 3: Market class implementation.



4. Functionality and alternatives

The main estimation functionality of diseq aims at disentangling demand and supply using various econometric assumptions about the market structure. The package implements five model classes. The equilibrium model (`equilibrium_model`) can be estimated with either two-stage least squares or full information maximum likelihood[3]. The package systemfit can be directly used with the option `2SLS` to obtain the estimates of the former methodology. The maximum likelihood estimation does not have alternative implementations. The basic disequilibrium model (`diseq_basic`) is also implemented by package the Disequilibrium. Disequilibrium estimates the basic model using `L-BFGS-B` via `optimr`, and the likelihood's gradient is numerically approximated. Instead, diseq allows the user to choose both the optimization method and whether numerical approximations or analytic gradient calculations are used. In addition, diseq provides options for using analytic expressions for

[3]See also table 3

the Hessian calculation when estimating standard errors. The remaining three disequilibrium provided models, namely the `diseq_directional`, `diseq_deterministic_adjustment`, and `diseq_stochastic_adjustment`, do not have implementation alternatives.

The estimation methods implemented by diseq have macroeconomic origins (Fair, 1971; Maddala & Nelson, 1974), but have found applications in recent research of economics and finance using microdata (see for instance (Bulligan et al., 2017) and (Carbó-Valverde et al., 2009)). In this respect, demand estimation methods such as the Almost Ideal Demand Systems (AIDS) of Deaton & Muellbauer (1980) and the structural estimation of Berry et al. (1995) (typically abbreviated as BLP) partially overlap with the methods described in this article. The AIDS and BLP methodologies are micro-founded, but they focus only on the demand side and are not concerned with the market-clearing assumption. Implementations of these methods can be correspondingly found in the packages BLPestimatoR and micEconAids.

4.1. Initialization options

Model object initialization is performed via the `new` keyword (see listing 6 as well as listing 3). An initialization call requires specifying after the intended initialed model class in order (i) the entity and time identification columns of the dataset, (i.o) the time column for the dynamic models, (ii) the quantity column, (iii) the price column, (iv) the demand specification, (v) the supply specification, (v.o) the price equation specification in the case of `diseq_stochastic_adjustment` `model`, and (vi) the data. Two optional keyword arguments control whether the shocks of the stochastic equations are correlated (`correlated_shocks` with default value equal to `TRUE`), and the verbosity level of output messages that operations on the initialized object print (`verbose` with default value equal to 0. The default verbosity level prints only errors; level 1 additionally prints warnings, 2 basic information, 3 verbose information, and 4 debug information.

The constructor prepares the model's variables using the passed specifications, which are expected to follow the syntax of `formula`. The demand variables are extracted by a formula that uses the quantity column (argument (ii)) on the left hand side and the demand specification (argument (iv)) on the right hand side. The supply variables are analogously constructed by arguments (ii) and (v). In the case of the `diseq_stochastic_adjustment` `model`, the price dynamics' variables are extracted using price specification. The price specification should contain only terms other than that of excess demand, which after all is not observed. The constructor implicitly adds the excess demand term. A call to the `new` returns an initialized `S4` market model object. Basic and extended information about the initialized object is correspondingly accessible via the `show` and `summary` methods. Listing 6 and an example of initializing and summarizing a market model.

Listing 6: Initializing and summarizing a market model.

```
1 model <- new(
2   "diseq_stochastic_adjustment",
```

```
  c("ID", "TREND"), "TREND", "HS", "RM",
  "RM + TREND + W  + CSHS + MONTH",
  "RM + TREND + W  + MA6DSF + MA3DHF + MONTH",
  "TREND + L2RM + L3RM  + L4RM",
  fair_houses() %>% dplyr::mutate(
    HS = log(HS),
    L1HS = dplyr::lag(HS),
    CSHS = cumsum(ifelse(is.na(L1HS), 0, L1HS)),
    L3RM = dplyr::lag(RM, 3), L4RM = dplyr::lag(RM, 4)),
  correlated_shocks = FALSE)
summary(model)
```

```
Stochastic Adjustment Model for Markets in Disequilibrium
  Demand Equation   : D_HS ~ D_RM + D_W + D_TREND + D_CSHS + D_MONTH
  Supply Equation   : S_HS ~ S_RM + S_W + S_TREND + S_MA6DSF + S_MA3DHF + S_MONTH
  Price Equation    : RM_DIFF ~ (D_HS - S_HS) + TREND + L2RM + L3RM
  Shocks            : Independent
  Nobs              : 128
  Sample Separation : Not Separated
  Quantity Var      : HS
  Price Var         : RM
  Key Var(s)        : ID, TREND
  Time Var          : TREND
```

The main task of the constructor is to prepare the data for estimation. Specifically:

i. If the passed data set contains rows with `NA` values, then these are dropped. If the verbosity level allows warnings, a warning reporting how many rows were dropped is emitted.

ii. After dropping the rows, factor levels may be invalidated. If needed, the constructor readjusts the factor variables by removing the unobserved levels. Factor indicators and interaction terms are automatically created according to the given specifications of the model's equations.

iii. A single primary key column is constructed by pasting the values of the entity and time key columns.

iv. In the cases of the `diseq_directional`, `diseq_deterministic_adjustment`, and the `diseq_stochastic_adjustment` models, a column with lagged prices is constructed. Since lagged prices are unavailable for the observations of the first time point, these observations are dropped. If the verbosity level allows the emission of information messages, the constructor prints the number of dropped observations.

v. In the cases of the `diseq_directional` and the `diseq_stochastic_adjustment` models, a column with price differences is created.

4.2. Estimation options

A market model object can be estimated using the `estimate` command. A call to `estimate` requires only passing an initialized model object. By default, the model is estimated by maximizing its likelihood using `BFGS` with analytically calculated gradient expressions. The standard errors are calculated using analytic Hessian expressions when these are available (`diseq_basic` and `diseq_directional`) and numerical approximations otherwise, and the standard errors are assumed to be homoscedastic. The default options can be overridden using the three available keyword arguments of the `estimate` methods.

The `gradient` keyword admits one of two potential values. Passing `"numerical"` instructs the `estimate` method to use numerically approximated gradient, while `"calculated"` to use analytic expressions. The `hessian` keyword can take one of three potential options. Passing `"skip"` does not calculate the Hessian (and hence standard errors), and the options `"numerical"` and `"calculated"` are analogous to the `gradient` options. The keyword argument `standard_errors` takes either one of the predefined values `"homoscedastic"` and `"heteroscedastic"` , or a vector with data variables names for which standard error clusters are to be created. If the option `"heteroscedastic"` is passed, the standard errors are calculated using the Huber-White heteroscedasticity adjustment, and the variance-covariance matrix is estimated using the sandwich estimator

$$\hat{\mathrm{Var}}_h(\hat{\theta}) = \left(\mathrm{H}(\hat{\theta})\right)^{-1} \mathrm{S}(\hat{\theta})'\mathrm{S}(\hat{\theta}) \left(\mathrm{H}(\hat{\theta})\right)^{-1},$$

where H is the Hessian and S the score matrix. If a vector of data variables is supplied, the variance-covariance matrix is calculated by grouping the score matrix based on the passed variables, i.e.

$$\hat{\mathrm{Var}}_c(\hat{\theta}) = \left(\mathrm{H}(\hat{\theta})\right)^{-1} \left(\sum_{j=1}^{M} \mathrm{S}_j(\hat{\theta})'\mathrm{S}_j(\hat{\theta})\right) \left(\mathrm{H}(\hat{\theta})\right)^{-1},$$

where M is number of clusters created by the distinct rows of the combinations of the columns corresponding to the given variable names and S_j is the score matrix containing only the likelihood contributions of the j-th cluster. Listing 7 gives an example of estimating the model of listing 6 using clustered standard errors.

Listing 7: A customized estimation call.

```
1 est <- estimate(model, control = list(maxit = 1e+5, reltol = 1e-4),
2                 standard_errors = c("W"))
3 bbmle::summary(est)
```

Additional keyword arguments given to `estimate` are passed down to the `bbmle::mle2`. The caller can customize the used optimization method, optimization options, and the estimations' starting values through

this. By default, the estimation uses the estimates of linear regressions of equations of the model's system to initialize the optimizer. For the case of the `equilibrium_model`, setting the `method` keyword argument equal to `"2SLS"` can be used to switch from maximum likelihood to two-stage least square estimation. Any additional keyword arguments are then passed down to a `systemfit::systemfit` call.

4.3. Post-estimation options

The post-estimation functionality of diseq can be categorized into three groups. These are (i) methods that return aggregated and dis-aggregated demanded and supplied quantities (i.e., predicted values), (ii) methods for the analysis for shortages, and (iii) methods for calculating marginal effects for market variables.

4.3.1 Demand, supplied quantities, and aggregation

The methods `demanded_quantities` and `supplied_quantities` take as argument a model object and a parameter vector and return the predicted values of the demand and supply equation at the given vector. The methods `aggregate_demand` and `aggregate_supply` offer basic default aggregation functionality. They also receive a model object and a parameter vector and calculate the sample's aggregate demand or supply at the passed set of parameters. If the model is static, as is the case of `equilibrium_model`, then all observations are aggregated. If the model has a dynamic component, such as the `diseq_stochastic_adjustment`, then demanded and supplied quantities are automatically aggregated within each time point. Listing 8 gives an example of calling all four methods using the estimated coefficients of the model object of listing 6. The output is omitted for brevity.

Listing 8: Predicted and aggregated quantities.

```
1 demanded_quantities(model, est@coef)
2 supplied_quantities(model, est@coef)
3 aggregate_demand(model, est@coef)
4 aggregate_supply(model, est@coef)
```

4.3.2 Analysis of shortages

Six methods can be used in the analysis of shortages. All methods expect an initialized model object and a vector of model parameters as input. Listing 9 offers an example of calling these methods for the model object of listing 6. The method `shortages` returns the difference of predicted demanded and supplied quantities, i.e.

$$\hat{XD} = \hat{D} - \hat{S}.$$

A call to `shortage_indicators` returns a vector with Boolean values indicating whether the model predicts a shortage (`TRUE`) or a surplus (`FALSE`) for the corresponding observation. The method `shortage_standard_deviation`

calculates the estimated shortage standard deviation by

$$\hat{\mathrm{Var}}(\hat{XD}) = \sqrt{\hat{\sigma}_d^2 + \hat{\sigma}_s^2 - 2\hat{\rho}\hat{\sigma}_d\hat{\sigma}_s}.$$

Normalized shortages are calculated by `normalized_shortages` according to

$$\hat{XD}_n = \frac{\hat{XD}}{\hat{\mathrm{Var}}(\hat{XD})}.$$

The normalization is common for all observations in the sample. Instead, the method `relative_shortages` idiosyncratically scales the shortages of the sample by the corresponding predicted supplied quantities, namely

$$\hat{XD}_r = \frac{\hat{XD}}{\hat{S}}.$$

Lastly, the method `shortage_probabilities` calculates and returns the probabilities

$$\hat{\pi_D} = \Phi(\hat{XD}_n),$$

where Φ denotes the standard normal distribution.

Listing 9: Shortage methods.

```
shortages(model, est@coef)
shortage_indicators(model, est@coef)
shortage_standard_deviation(model, est@coef)
normalized_shortages(model, est@coef)
relative_shortages(model, est@coef)
shortage_probabilities(model, est@coef)
```

4.3.3 Marginal effects

The last functionality group of diseq concerns the calculation of marginal effects. Listing 10 exemplifies the marginal effect functionality by calculating them for the object model of listing 6. For a `shortage_marginal` call, besides specifying a model object and a parameter vector, one needs to specify the variable name for which the marginal system effect is calculated. The variable name should be given to the method without any equation prefix. The method returns

$$M_x = \frac{1}{\hat{\mathrm{Var}}(\hat{XD})}\frac{\partial \hat{XD}}{\partial x}.$$

If the variable is found in both demand and supply sides, the returned value is named by prefixing the base variable name with `"B"`. Otherwise, the result is prefixed by `"D"` (`"S"`) if it is only found on the demand (supply) side. The system marginal effect M_x takes into account both sides of the market, but it is constant with respect

to the coefficients of the demand and supply variable because demand and supply are linear functions of these coefficients.

A state-dependent evaluation of the effect of available on the system can be obtained by calculating the marginal effect on the probability of observing a shortage. The method `shortage_probability_marginal` expects the same type of input arguments with `shortage_marginal`. In addition, it accepts an optional keyword argument `aggregate` that controls how averaging over states is performed. When the `aggregate` is set to `"mean"` (the default value), the method returns mean marginal effects, which are calculated by

$$P_x = \mathbb{E}\,\frac{\partial \hat{\pi_D}}{\partial x} = M_x\,\mathbb{E}\,\varphi(\hat{XD}),$$

where φ is the standard normal density. Marginal effects at the mean are calculated by

$$\tilde{P}_x = M_x\,\varphi(\mathbb{E}\,\hat{XD}),$$

when `aggregate` is set to `"at_the_mean"`.

Listing 10: Marginal effects.

```
1 shortage_probability_marginal(model, est@coef, "RM")
2 shortage_probability_marginal(model, est@coef, "CSHS", aggregate = "at_the_mean")
3 shortage_marginal(model, est@coef, "MA3DHF")
```

```
>     B_RM
-0.0008854701
>     D_CSHS
0.02019373
>     S_MA3DHF
-0.001261552
```

5. Estimation benchmarks

A major difficulty in estimating models for markets in disequilibrium comes from their computational complexity. Dorsey & Mayer (1995) classify the estimation of disequilibrium models among the most demanding econometric estimation problems, as the likelihoods of these models have poles and non-unique local maxima. In addition, the authors propose genetic algorithm optimization methods for estimating the basic model and compare its computational performance with Nelder-Mead. As proposed by Maddala (1986), the classic estimation approach obtains the maximum likelihood estimates using a global, iterative Newton method. Zilinskas & Bogle (2006) use random interval arithmetic optimization for locating global maxima. They apply the technique to the basic model with independent shocks using the dataset of Fair & Jaffee (1972) to assess its performance experimentally.

Bowden (1978) considers the deterministic and stochastic adjustment models and proposes a re-parametrization that allows them to be estimated using more straightforward procedures. Instead, Quandt & Ramsey (1978) estimate the stochastic adjustment model with a methodology based on the moment generating function of the likelihood.

This section compares the maximum likelihood estimation performance in terms of computation time for the three optimization options that diseq provides. It examines the mean estimation time of likelihood maximization via `BFGS` with analytically calculated gradients, `BFGS` with numerically approximated gradients, and the simplex method of Nelder-Mead. Statistics on the execution times are collected via a series of benchmarking simulations performed in the CSC cluster of Goethe University.

The models are simulated using both random parameters and samples. The independent and explanatory variables of each model are drawn from normal distributions. The sample data are then generated using the structure of the model and the randomly drawn parameters. Estimating the models using `BFGS` with numerically approximated gradients is the most error-prone procedure among the three examined algorithms because numerical differentiation fails in many occasions nearby poles. To ensure that the collected statistics accurately measure execution times and are not biased from estimation failures, an untimed estimation using `BFGS` with numerically approximated gradients is executed for each set of simulated parameters and sample. If the estimation succeeds, the generated data are used in timed executions, and a new data sample is generated otherwise.

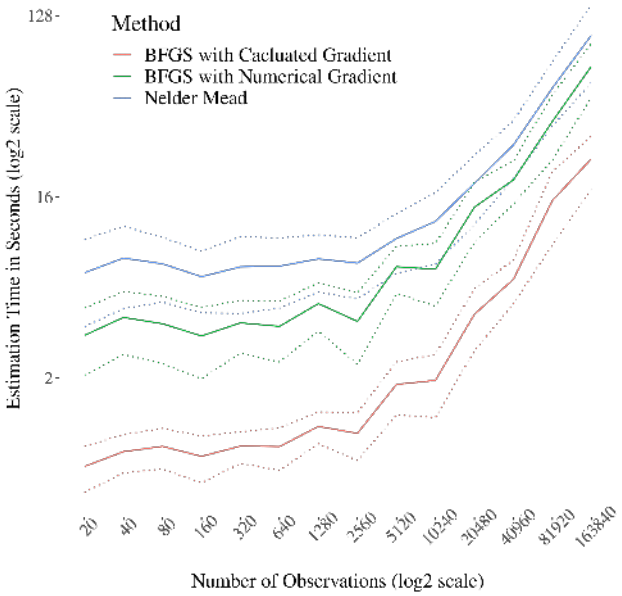
Each model is simulated 100 times for a set of 14 different sample sizes and 14 different sets of model parameters, and it is ensured that the simulated data are economically well behaved. If prices or quantities are negative, or if shortages and surpluses represent more than 90% of the sample, the simulation round is repeated. To keep things equal, each simulated data set is used to estimate the model with all three optimization tools. The execution time is saved at the end of each round. The counted time concerns only the estimation of the models and not their simulation or the calculations of standard errors. The estimation tolerance is kept constant for all optimization methods. To spin up the processors, 2 untimed warm-up estimations are performed in the beginning.

Appendix A details the data generating process of each model. Diseq exposes this simulation functionality via the functions `simulate_data` and `simulate_model`. The results are depicted in figures 4 (`equilibrium_model`), 5 (`diseq_basic`), 6 (`diseq_directional`), 7 (`diseq_deterministic_adjustment`), and 8 (`diseq_stochastic_adjustment`). The vertical axes of the figures measure the mean estimation time of each optimization method. The horizontal axes of figs. 4a, 5a, 6a, 7a and 8a measure number of observations of the simulated sample for a constant number of simulated parameters[4]. The horizontal axes of figs. 4b, 5b, 6b, 7b and 8b measure number simulated

[4]The number of parameters depends on the simulated model. The equilibrium and basic models use 14 parameters, namely 6 demand parameters, 5 supply parameters, 2 variances, and a correlation. The directional model uses 13 parameters because prices cannot be used in both sides of the market. The deterministic adjustment model uses 15 parameters since it introduces

Figure 4: Equilibrium model execution time benchmarks.

(a) Over observations.



(b) Over parameters.

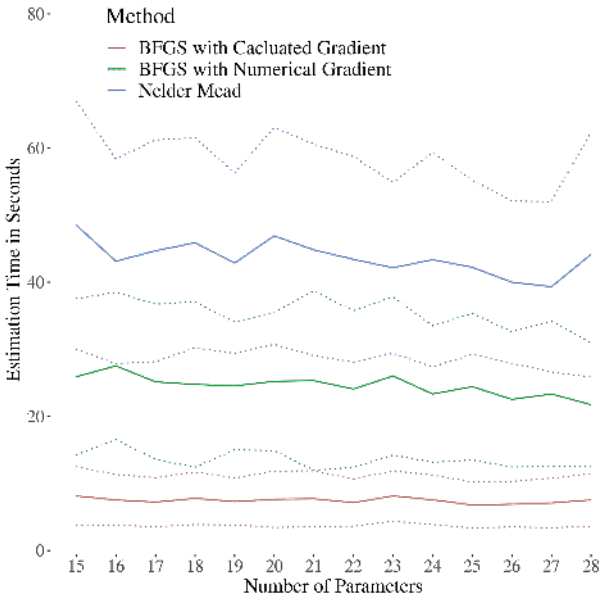
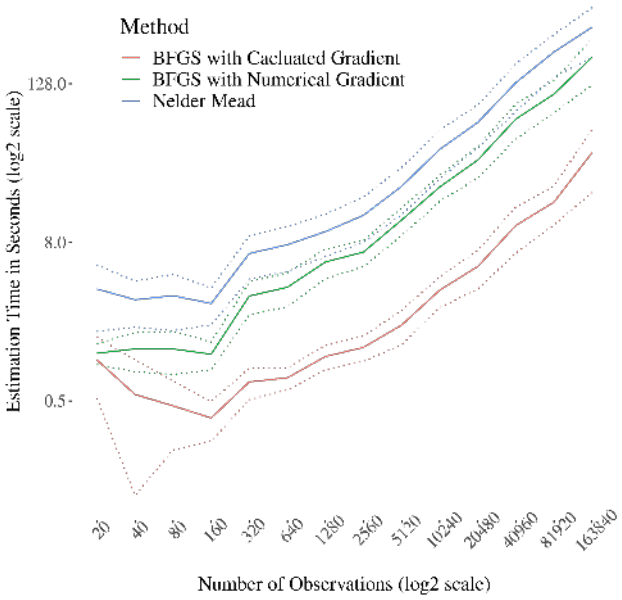
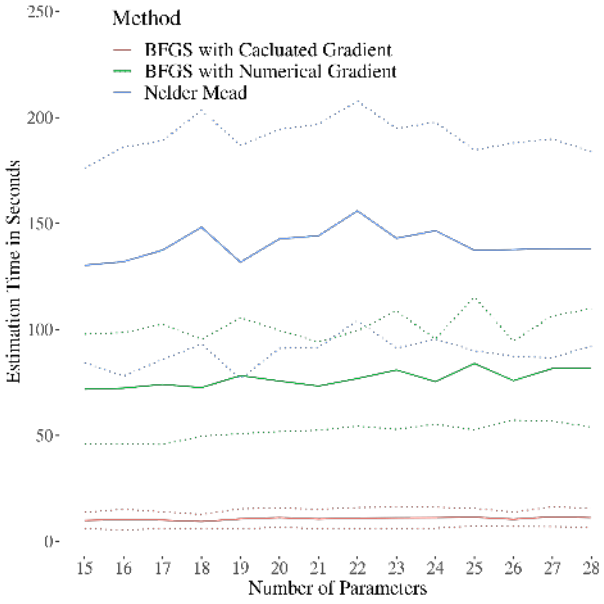


Figure 5: Basic model execution time benchmarks.

(a) Over observations.



(b) Over parameters.



an additional parameter in the price dynamics. Lastly, the stochastic adjustment model uses 20 parameters, three of which are introduced in the price dynamics, one additional variance, and two correlations.

Figure 6: Directional model execution time benchmarks.

(a) Over observations.

(b) Over parameters.

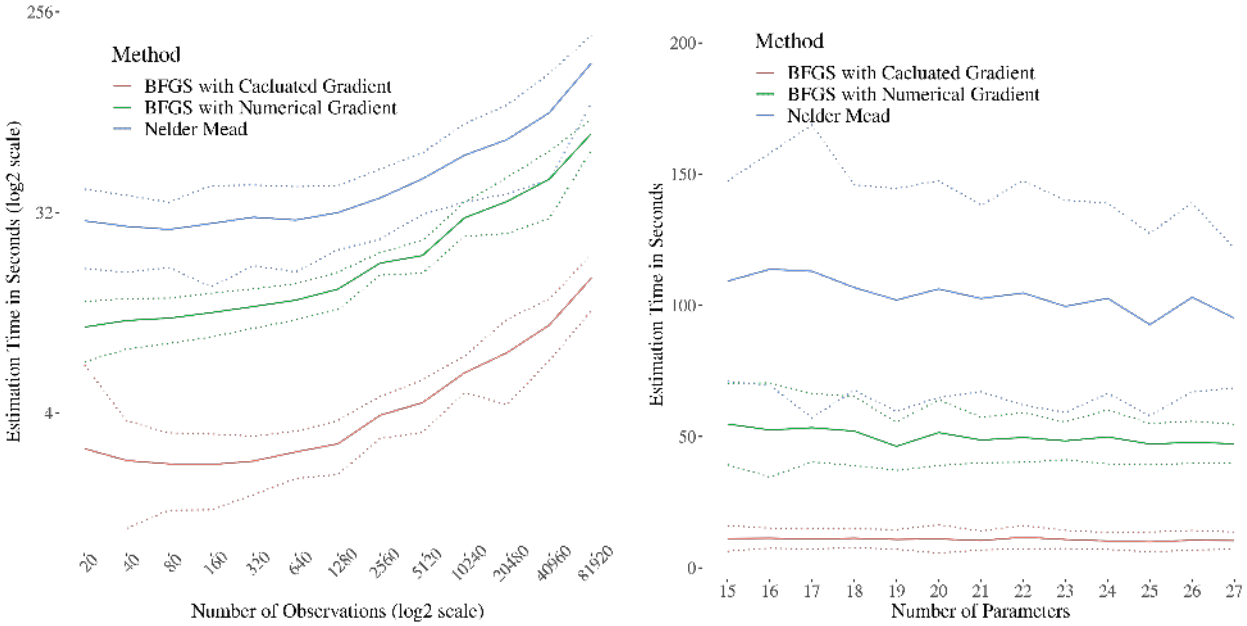


Figure 7: Deterministic adjustment model execution time benchmarks.

(a) Over observations.

(b) Over parameters.

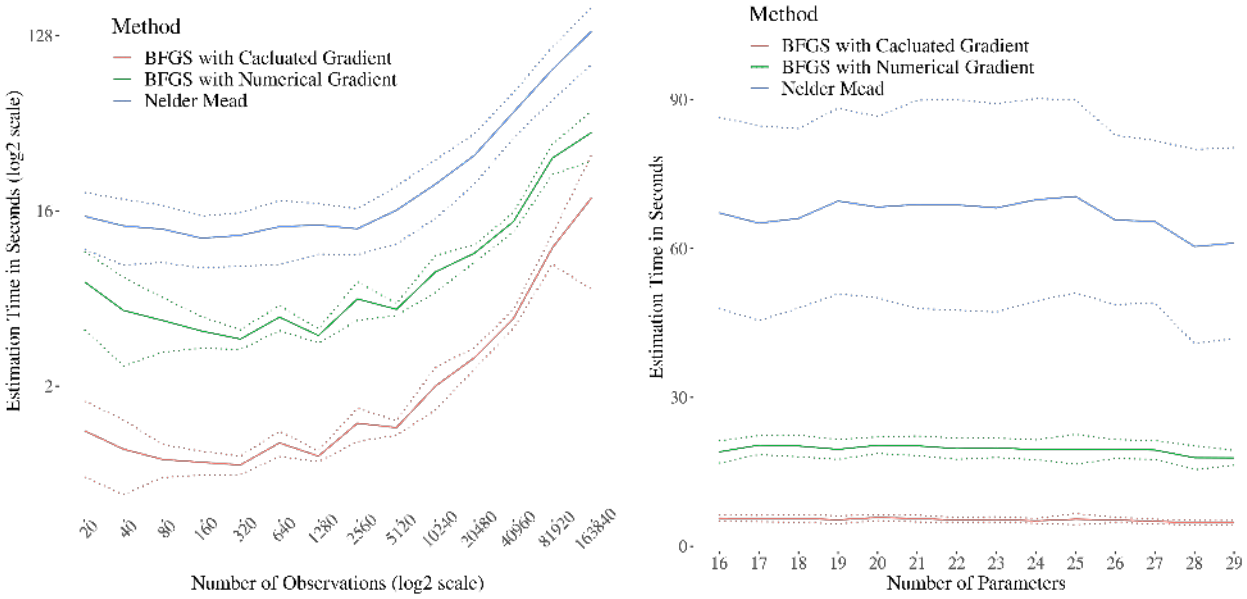
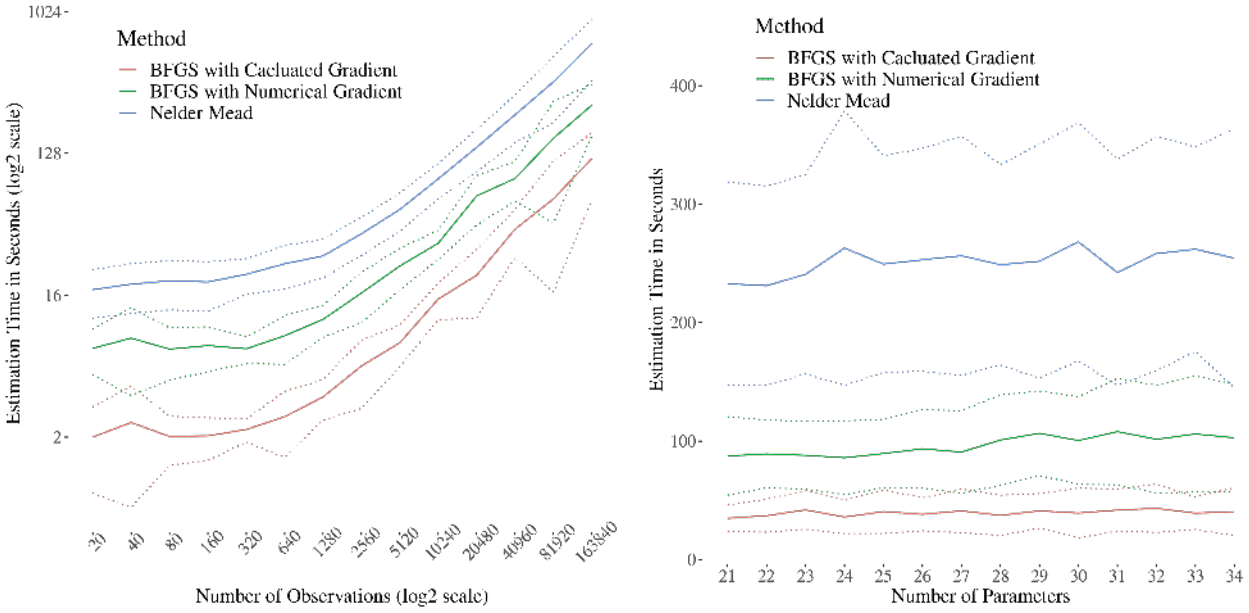


Figure 8: Stochastic adjustment model execution time benchmarks.

(a) Over observations.

(b) Over parameters.



parameters for a constant sample size of 40,960 observations. The points of solid lines represent mean execution times over 100 estimations, and dotted lines depict 1 standard deviation intervals from the measured means.

The results exhibit similar patterns for all five models. The estimation time grows exponentially in the size of the sample. Instead, small changes in the number of estimated parameters do not significantly affect estimation times. Out of the three compared methods, `Nelder-Mead` results on average to the lengthiest estimation times and the greatest execution time variability. In all cases, `BFGS` with analytically calculated gradients is the most efficient estimation option among those available in diseq. For instance, the basic model, which is the most used among all the disequilibrium models, is estimated on average at least 6.43 times faster when the analytic expressions are used than when the gradient is numerically approximated (10.73 to 69.07 seconds correspondingly).

6. Conclusion

This article has introduced the diseq R package. The package provides methods for estimating, simulating, and analyzing markets in equilibrium and disequilibrium. The article offers a guided tour to the core functionality of the five market models implemented in diseq via an overarching empirical example using the classic dataset of Fair & Jaffee (1972) and details the remaining methods of the package using standalone examples.

The estimation functionality of diseq is based on analytic gradient and Hessian expressions of the likelihoods

of the implemented market models, which are not available in other statistical software. This implementation characteristic attributes a computational edge to diseq in terms of estimation time efficiency. The efficiency gains are documented by gathering and presenting statistics of estimation performance in large-scale benchmarking simulations.

Acknowledgments

I had a series of motivating discussions regarding the functionality of the package with Oivind Nilsen. Alexander Ludwig has given me access to the CSC cluster of Goethe University, which I have used for the computationally intensive parts of the development process. I have also received helpful comments while presenting various applications and results using the functionality of the diseq from participants of the conference 'Modelling with Big Data and Machine Learning: Measuring Economic Instability', the Goethe University's Management and Microeconomics Colloquium of 2020, and the Brown Bag seminar of the Leibniz Institute SAFE. Various members of the CRAN project guided me through the process of making the package available. I am thankful to all.

References

Balestra, P., & Varadharajan-Krishnakumar, J. (1987). Full information estimations of a system of simultaneous equations with error component structure. *Econometric Theory*, *3*(2), 223–246. doi: 10.1017/S0266466600010318

Bbmle. (2020). *Tools for General Maximum Likelihood Estimation.* By Bolker B, R Development Core Team & Giné-Vázquez I. Distributed by CRAN. Retrieved from https://cran.r-project.org/package=bbmle

Berry, S., Levinsohn, J., & Pakes, A. (1995). Automobile Prices in Market Equilibrium. *Econometrica*, *63*(4), 841–890.

BLPestimatoR. (2019). *BLP Demand Estimation.* By Brunner D., Weiser C. & Romahn A. Distributed by CRAN. Retrieved from https://cran.r-project.org/web/packages/BLPestimatoR/index.html

Bowden, R. J. (1978). Specification, Estimation and Inference for Models of Markets in Disequilibrium. *International Economic Review*, *19*(3), 711. doi: 10.2307/2526335

Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, *6*(1), 76–90. doi: 10.1093/imamat/6.1.76

Bulligan, G., Busetti, F., Caivano, M., Cova, P., Fantino, D., Locarno, A., & Rodano, M. L. (2017, jul). *The Bank of Italy Econometric Model: An Update of the Main Equations and Model Elasticities* (Temi

di discussione (Economic working papers) No. 1130). Bank of Italy, Economic Research and International Relations Area. doi: 10.2139/ssrn.3040651

Carbó-Valverde, S., Rodrguez-Fernndez, F., & Udell, G. F. (2009). Bank market power and SME financing constraints. *Review of Finance*, *13*(2), 309–340. doi: 10.1093/rof/rfp003

Deaton, A., & Muellbauer, J. (1980). An Almost Ideal Demand System. *The American Economic Review*, *70*(3), 312–326.

Diseq. (2021). *Estimation Methods for Markets in Equilibrium and Disequilibrium.* Distributed by CRAN. Retrieved from https://cran.r-project.org/package=diseq

Disequilibrium. (2020). *Disequilibrium Models.* By Latshaw N. & Guggisberg M. Distributed by CRAN. Retrieved from https://cran.r-project.org/package=Disequilibrium

Dorsey, R. E., & Mayer, W. J. (1995). Genetic algorithms for estimation problems with multiple optima, nondifferentiabiiity, and other irregular features. *Journal of Business and Economic Statistics*, *13*(1), 53–66. doi: 10.1080/07350015.1995.10524579

Economic Report of the President (Tech. Rep.). (1947). United States: President and Council of Economic Advisers (U.S.). Retrieved from https://fraser.stlouisfed.org/title/45

Fair, R. C. (1971). *A short-run forecasting model of the United States economy.* Heath Lexington Books. Retrieved from https://fairmodel.econ.yale.edu/RAYFAIR/pdf/1971EI.PDF

Fair, R. C., & Jaffee, D. M. (1972). Methods of Estimation for Markets in Disequilibrium. *Econometrica*, *40*(3), 497. doi: 10.2307/1913181

Federal Reserve Bulletin (Tech. Rep.). (1914). Board of Governors of the Federal Reserve System (U.S.), 1935- and Federal Reserve Board. Retrieved from https://fraser.stlouisfed.org/title/62

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, *13*(3), 317–322. doi: 10.1093/comjnl/13.3.317

Goldfarb, D. (1970). A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation*, *24*(109), 23–26.

Henningsen, A., & Hamann, J. D. (2007). systemfit : A Package for Estimating Systems of Simultaneous Equations in R. *Journal of Statistical Software*, *23*(4), 1–40. Retrieved from http://www.jstatsoft.org/v23/i04/ doi: 10.18637/jss.v023.i04

Hwang, H.-s. (1980). A test of a disequilibrium model. *Journal of Econometrics*, *12*(3), 319–333. doi: 10.1016/0304-4076(80)90059-7

Maddala, G. S. (1986). Disequilibrium, self-selection, and switching models. In *Handbook of econometrics* (Vol. 3, pp. 1633–1688). Elsevier. doi: https://doi.org/10.1016/S1573-4412(86)03008-8

Maddala, G. S., & Nelson, F. D. (1974). Maximum Likelihood Methods for Models of Markets in Disequilibrium. *Econometrica*, *42*(6), 1013. doi: 10.2307/1914215

MicEconAids. (2017). *Demand analysis with the almost ideal demand system.* By Henningsen A. Distributed by CRAN. Retrieved from `https://cran.r-project.org/package=micEconAids`

Quandt, R. E., & Ramsey, J. B. (1978). Estimating mixtures of normal distributions and switching regressions. *Journal of the American Statistical Association*, *73*(364), 730–738. doi: 10.1080/01621459.1978.10480085

Shanno, D. (1970). Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, *24*(111), 647–656.

Zellner, A., & Theil, H. (1962). Three-Stage Least Squares: Simultaneous Estimation of Simultaneous Equations. *Econometrica*, *30*(1), 54. doi: 10.2307/1911287

Zilinskas, J., & Bogle, I. D. L. (2006). Balanced random interval arithmetic in market model estimation. *European Journal of Operational Research*, *175*(3), 1367–1378. doi: 10.1016/j.ejor.2005.02.013

A. Simulation Details

Simulation methods are available for all five models provided by diseq. Two simulation options are available for each market model. The user may choose to generate a dataset based on the stochastic process implied by the model with and without initializing a model object. These two options are accessed by correspondingly calling the methods `simulate_model` and `simulate_data`. Essential, the first method is a wrapper of the second method combined with a constructor call. The first method is more helpful when only one model is to be examined, while the second when the same data are to be used for comparing multiple models. The simulation functionality is used in the unit tests of diseq, the documentation examples and vignettes, and the benchmarking exercise of this article.

All simulations begin with the same baseline specifications for demand and supply, which are given by eqs. (1) and (2). In the equilibrium case, prices are not simulated. Instead, they are calculated so that the market clears, i.e.

$$P_{n,t} = \frac{\sum_{j=1}^{k} \left(\eta_j^s - \eta_j^d\right) X_{n,t}^j + \beta_0^s - \beta_0^d + \sum_{j=1}^{k_s} \beta_j^s X_{n,t}^{j,s} - \sum_{j=1}^{k_d} \beta_j^d X_{n,t}^{j,d} + u_{n,t}^s - u_{n,t}^d}{\alpha^d - \alpha^s}.$$

In the basic model case, prices are simulated as a common control. The demanded and supplied quantities are then calculated, and the observed quantity is determined by the short side rule (that is eq. (4)). In the

directional model, prices are also simulated as the rest of the controls, and it is subsequently checked whether the condition $\Delta P_{nt} \geq 0 \implies D_{nt} \geq S_{nt}$ is true. If not, the simulation of controls and prices is repeated until a sample draw that fulfills the condition is realized. In the deterministic adjustment model, an out of sample, initial price value is drawn, and the remaining prices are then generated by the rule

$$P_{n,t} = \frac{\sum_{j=1}^{k} \left(\eta_j^s - \eta_j^d\right) X_{n,t}^j + \beta_0^s - \beta_0^d + \sum_{j=1}^{k_s} \beta_j^s X_{n,t}^{j,s} - \sum_{j=1}^{k_d} \beta_j^d X_{n,t}^{j,d} - \gamma P_{n,t-1} + u_{n,t}^s - u_{n,t}^d}{\alpha^d - \alpha^s - \gamma}.$$

For the stochastic adjustment model, the procedure is analogous to the one with deterministic price dynamics, but the price generation rule is now given by

$$P_{n,t} = \frac{\left(\eta_j^s - \eta_j^d\right) X_{n,t}^j + \beta_0^s - \beta_0^d + \beta_j^s X_{n,t}^{j,s} - \beta_j^d X_{n,t}^{j,d} - \gamma P_{n,t-1} + \gamma\beta_0^p + \gamma\beta_j^p X_{n,t}^{j,p} + u_{n,t}^s - u_{n,t}^d + \gamma u_{n,t}^p}{\alpha^d - \alpha^s - \gamma},$$

where Einstein summation notation over j is used to save some space. The simulation methods perform various validity checks in the generated data and instruct the user to reparametrize the model if any of them fails. For instance, it is ensured that simulated quantities and prices are positive, and that the sample is balanced between demand and supply observations.

A call to `simulate_data` requires specifying the model to be simulation as the first argument by passing the model string as it used in initialization calls (see listing 5 for an simulation call example). The model separates are specifying by keyword arguments to the function call. These keywords are `nobs`, `tobs`, `alpha_d`, `beta_d0`, `beta_d`, `eta_d`, `alpha_s`, `beta_s0`, `beta_s`, `eta_s`, `gamma`, `beta_p0`, `beta_p`, `sigma_d`, `sigma_s`, `sigma_p`, `rho_ds`, `rho_dp`, `rho_sp`. The names of the keywords follow the notation of this article, and correspond to the symbols of eqs. (1), (2), (4), (6) and (7). The default values of the variances is one, and of the correlations is zero. The caller can optionally also pass values the `seed`, `verbose`, `price_generator`, and `control_generator` input arguments. The last two options expect a function callback that is given an integer n and returns n randomly generated values. The default generators return normally distributed values with mean 2.5 and standard deviation 0.5.

The `simulate_model` call follows a similar calling convention. The difference is that the parameter and generator arguments that are intended to be passed down to `simulate_data` have to be specified as a list through the `simulation_parameters` input variable. The `seed` and the `verbose` keywords should be specified separately from the `simulation_parameters` list. Any additional keyword arguments given to `simulate_model` are passed down to the model's construction call.