This note describes the content of the MATLAB codes for the paper "Search Frictions and Efficiency in Decentralized Transport Markets". The data employed in this paper can be accessed via Clarksons Research and exactEarth Ltd. Since the data and part of the estimation procedure are the same as in "Geography, Transportation, and Endogenous Trade Cost,", Econometrica, 88(2),657-691, the codes to implement the estimation of the matching function and ships' costs and are included in the replication package there. Here, we focus on the codes to estimate the exporter side parameters, and to compute the counterfactuals.

Estimation

- `Estimation_exporters.m`: This file performs the estimation of exporter side parameters.
  This file begins by loading the estimates for ship costs as well as exporters' revenues from trading.
  Next, we estimate the exporters' inventory costs and bargaining parameters matching simulated and observed trading prices.
  Finally, we estimate the exporter entry cost, which captures both the cost of production, as well as any export costs beyond shipping prices. To reduce noise, we cluster the entry costs into n_{gp} groups.
- `obj.m`: This function computes the sum of squared differences between observed and predicted prices.

Platform

- `Platform allocation.m`: This file contains the codes to compute the counterfactual equilibrium allocation if the market was served by a centralizing platform. The platform eliminates search frictions through centralization and maximizes its profits.
  We begin by inserting the estimates of ships' and exporters' costs as well as the estimates of the matching function. Then, we proceed to simulate the allocation, based on the algorithm described in detail in Online Appendix C4.
- `Dprofits_dq.m`: This function computes the derivative of the platform's profits with respect to the number q_{ij} of carriers traveling full. See Lemma 6 in Appendix C4 for the specification.
- `Dprofits_ds.m`: This function computes the derivative of the platform's profits with respect to the number s_i of carriers waiting. See Lemma 6 in Appendix C4 for the specification.
- `entrants.m`: This function computes the number of entrants e_{ij} that sustain the pair (q_{ij}, s_i)
- `prices.m`: This function computes the prices p_{ij} that ensure that exactly q_{ij} exporters demand transportation on route ij
- `inner_loop.m`: This function computes the average payoff δ and ballast probabilities $\mathbb{P}_{ij}^b$ that implement (q_{ij}, s_i)
- `Welfare.m`: This function computes the total welfare for a given allocation.

- `ccp_equil.m`: This function computes the ships' choice probabilities.

Constrained efficient allocation

- `constrained_efficient_allocation.m`: This file contains the codes to compute the counterfactual equilibrium allocation if market prices coincided with the efficient prices.
  We begin by inserting the estimates of ships' and exporters' costs as well as the estimates of the matching function. Then, we proceed to simulate the allocation, based on the algorithm described in detail in Online Appendix C3.

- `efficient_equil.m`: This function computes the constrained efficient allocation iterating over the number of ships and carriers searching in each region. Prices are set to coincide with the efficient one (see equation 31 in the paper).

- `Welfare.m`: This function computes the total welfare for a given allocation.

- `ccp_equil.m`: This function computes the ships' choice probabilities.

Market allocation

- `market_allocation.m`: This file contains the codes to compute the market allocation under Nash bargaining.
  We begin by inserting the estimates of ships' and exporters' costs as well as the estimates of the matching function. Then, we proceed to compute the equilibrium allocation by iterating over the number of ships and exporters waiting at each port.

- `alg_1.m`: Given meeting rates, this function computes the ships' and exporters' value functions, as well as bargained prices, and exporters' choice probabilities.

- `Welfare.m`: This function computes the total welfare for a given allocation.

- `ccp_equil.m`: This function computes the ships' choice probabilities.

Frictionless allocation

- `Frictionless allocation.m`: This file contains the codes to compute the counterfactual equilibrium allocation if search frictions were eliminated.
  We begin by inserting the estimates of ships' and exporters' costs as well as the estimates of the matching function. Then, we proceed to simulate the allocation, based on the algorithm described in detail in Online Appendix C4.

- `Dwelfare_dq.m`: This function computes the derivative of welfare with respect to the number q_{ij} of carriers traveling full. See Lemma 6 in Appendix C4 for the specification.

- `Dprofits_ds.m`: This function computes the derivative of welfare with respect to the number s_i of carriers waiting. See Lemma 6 in Appendix C4 for the specification.
- `entrants.m`: This function computes the number of entrants e_{ij} that sustain the pair (q_{ij}, s_i).
- `prices.m`: This function computes the prices p_{ij} that ensure that exactly q_{ij} exporters demand transportation on route ij.
- `inner_loop.m`: This function computes the average payoff δ and ballast probabilities $\mathbb{P}_{ij}^b$ that implement (q_{ij}, s_i).
- `Welfare.m`: This function computes the total welfare for a given allocation.
- `ccp_equil.m`: This function computes the ships' choice probabilities.