Optimal Execution with Non-linear Market Impact

ACTSC972 Final Project

Bill Zhuo
University of Waterloo

December 4, 2021

Agenda

1 Continuous Propagator Model for Nonlinear Market Impact

2 Homotopy Analysis Method

3 Numerical Nonlinear Programming

4 Empirical Tests

5 Possible Extensions

Continuous Propagator Model for Nonlinear Market Impact

Propagator Models

Discrete Propagator Model (Bouchaud 2004)[1]

$$S_j = S_0 + \sum_{k=0}^{j-1} \left(\mathcal{G}(j, k, v_k) + \eta_k \right), \quad \mathcal{G}(j, k, v_k) \cong f(v_k) \cdot G(j, k)$$

Propagator Models

Discrete Propagator Model (Bouchaud 2004)[1]

$$S_j = S_0 + \sum_{k=0}^{j-1} \left(\mathcal{G}(j, k, v_k) + \eta_k\right), \quad \mathcal{G}(j, k, v_k) \cong f(v_k) \cdot G(j, k)$$

Continuous Propagator Model (Gatheral 2009)[5]

$$S_t = S_0 + \int_0^t f(v_s) G(t-s) ds + \int_0^t \sigma dZ_s$$

Some Definitions

- $f(\cdot)$: (instantaneous) market impact function
- $G(\cdot)$: decay kernel
- Π: an execution strategy/scheme

$$\int_0^T v(t)dt = X$$

where X is the amount of shares to execute.

Some Definitions

- $f(\cdot)$: (instantaneous) market impact function
- $G(\cdot)$: decay kernel
- Π: an execution strategy/scheme

$$\int_0^T v(t)dt = X$$

where X is the amount of shares to execute.

Expected Cost Function $C(\Pi)$

$$C(\Pi) = \mathbb{E}\left[\int_0^T v(t)(S_t - S_0)dt\right] = \int_0^T v(t) \int_0^t f(v(s))G(t-s)dsdt$$

also known as **expected implementation shortfall**.

A Constrained Optimization Problem

We want to find such v that solves the following problem

$$\begin{array}{ll} \min & C(\Pi) \\ \text{s.t.} & \int_0^T v(t)dt = X \end{array}$$

A Constrained Optimization Problem

We want to find such v that solves the following problem

$$\begin{array}{ll} \min & C(\Pi) \\ \text{s.t.} & \int_0^T v(t)dt = X \end{array}$$

Sign requirement on $v(t)$?

A Constrained Optimization Problem

We want to find such v that solves the following problem

$$\begin{array}{ll} \min & C(\Pi) \\ \text{s.t.} & \int_0^T v(t)dt = X \end{array}$$

Sign requirement on $v(t)$?

Consistency requirement on f and G?

A Constrained Optimization Problem

We want to find such v that solves the following problem

$$\begin{array}{ll} \text{min} & C(\Pi) \\ \text{s.t.} & \int_0^T v(t)dt = X \end{array}$$

Sign requirement on $v(t)$**?**

Consistency requirement on f **and** G**?**

Principle of no-dynamic-arbitrage: price manipulation is not possible.

$$\nexists \Pi \;\; \text{s.t.} \;\; C(\Pi) < 0$$

Why Consider Nonlinear *f*, *G*?

- Empirical results on *f*: power-law [9] or log impact function [1]

Why Consider Nonlinear f, G?

- Empirical results on f: power-law [9] or log impact function [1]
- Observed autocorrelation of trade signs [1] motivates a decay model

$$G(t-s) = (t-s)^{-\gamma}, \quad \gamma > 0$$

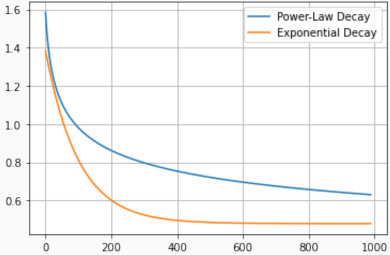


Figure: Power law vs. Exponential

Why Not Exponential Decay?

Obizhaeva and Wang 2005 [11]

Assume $f(x) \propto x$ and $G(t-s) = \exp\{-\rho(t-s)\}, \rho > 0$.

Why Not Exponential Decay?

Obizhaeva and Wang 2005 [11]

Assume $f(x) \propto x$ and $G(t-s) = \exp\{-\rho(t-s)\}, \rho > 0$.

Lemma (Gatheral[5])

If G is an exponential decay kernel, price manipulation is possible unless $f(x) \propto x$.

Clean Linear Case f

Assume f to be linear with the proposed G

Theorem (Gatheral, Schied[6])

$$v(t) = \frac{c}{[t(T-t)]^{\frac{1-\gamma}{2}}}$$

where

$$c = X / \left(\sqrt{\pi} \left(\frac{T}{2} \right)^{\gamma} \frac{\Gamma((1+\gamma)/2)}{\Gamma(1+\gamma/2)} \right)$$

This would be a benchmark solution called **GSS**.

Symmetric Execution

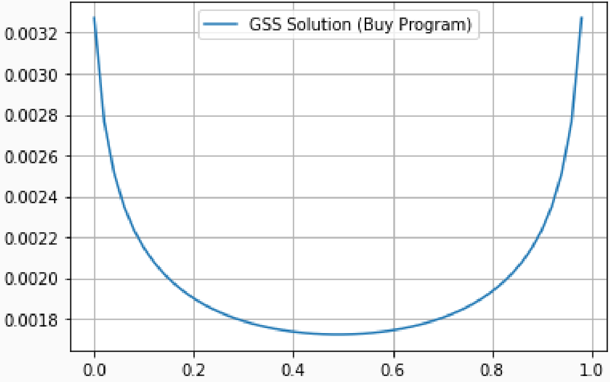


Figure: Linear impact function solution $v(t)$ with $T = 1$ and $X = 0.1, \gamma = 0.5$

Our Nonlinear Impact Function f

Nonlinear Impact Function f

We shall work with the following f

$$f(x) = \text{sign}(x)|x|^{\delta}, \delta > 0$$

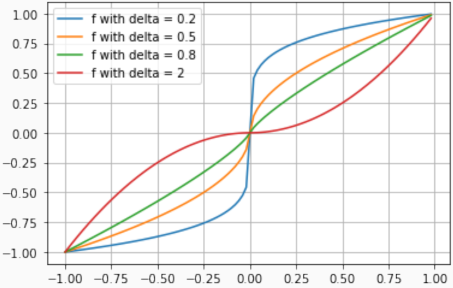


Figure: Impact function f with various $\delta > 0$ gives different nonlinearity properties

Our Problem to Solve

Theorem (Dang 2014[4])

The expected cost minimization problem is equivalent to solving the integral equation

$$\int_0^t f(v(s))G(t-s)ds + f'(v(t)) \int_t^T v(s)G(s-t)ds = \lambda$$

where λ is derived based on the constraint.

Our Problem to Solve

Theorem (Dang 2014[4])

The expected cost minimization problem is equivalent to solving the integral equation

$$\int_0^t f(v(s))G(t-s)ds + f'(v(t))\int_t^T v(s)G(s-t)ds = \lambda$$

where λ is derived based on the constraint.

Simplification

$$\int_0^T G(|t-s|)F(v(s),v(t))ds = \lambda$$

where $F(v(s),v(t)) = \begin{cases} f(v(s)) & s \leq t \\ v(s)f'(v(t)) & s > t \end{cases}$

What δ, γ to consider?

Theorem (Gatheral[5])

Dynamic-no-arbitrage imposes that

$$\gamma + \delta \geq 1$$

Homotopy Analysis Method

Homotopy Analysis Method (Liao)[8]

General Nonlinear Equation

$$\mathcal{N}[v(t)] = 0$$

Zero-Order Deformation Equation

$$(1-p)\mathcal{L}[\phi(t;p) - v^0(t)] = p\hbar H(t)\mathcal{N}[\phi(t;p)]$$

- $p \in [0,1]$ homotopy parameter *
- $\hbar \neq 0$ convergence control *
- $H(t)$ auxiliary function, $H(t) = 1$
- $\mathcal{L}$ linear operator $\mathcal{L} =$identity

We let $\mathcal{N}[v(t)] = -\lambda + \int_0^T G(|t-s|)F(v(s), v(t))ds$
Note that $\phi(t;0) = v^0(t)$ and $\phi(t;1) = v(t)$

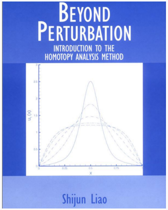


Figure: Cover page[7]

Homotopy Series and Homotopy Derivatives

Consider the Maclaurin series of ϕ with respect to p,

$$\phi(t;p) = v^0(t) + \sum_{m=1}^{\infty} v^m(t)p^m, \quad v^m(t) = \frac{1}{m!}\frac{\partial^m \phi(t;p)}{\partial p^m}\bigg|_{p=0}$$

Since $\phi(t;1) = v(t)$, we have

$$v(t) = v^0(t) + \sum_{m=1}^{\infty} v^m(t)$$

We don't really know $\phi(t;p)$, how to compute $v^m(t)$?

*m*th-order Deformation Trick

We can take partials on the zero-order deformation equation iteratively to get the *m*th-order deformation equation

$$v^m(t) - v^{m-1}(t) = \hbar R^m, \quad R^m = \frac{1}{(m-1)!} \frac{\partial^{m-1} \mathcal{N}[\phi(t;p)]}{\partial p^{m-1}}\bigg|_{p=0}$$

This tells us

$$v^m(t) = v^{m-1}(t) + \hbar R^m$$

Approximate Solution

Theorem (Approximate Solution Convergence (Liao 1997)[8])

The nth-order approximate solution is

$$v^{(n)}(t) = v^{0}(t) + \sum_{m=1}^{n} v^{m}(t)$$

and

$$\lim_{n\to\infty} v^{(n)}(t) = v(t)$$

Discrete Homotopy Analysis Method

DHAM

Consider N subintervals of $[0, T]$ with $t_i = iT/N$. A discrete solution is $v_i = v(t_i)$ for $i \in \{1, \cdots, N\}$.

$$\int_0^T G(|t-s|)F(v(s), v(t))ds = \lambda \longrightarrow \sum_{j=1}^{N} G_{ij}F_{ij}(v) = \lambda$$

with

$$F_{ij}(v) = \begin{cases} f(v_j) & j \leq i \\ v_j f'(v_i) & j > i \end{cases}, \quad G_{ij} = \int_{t_{i-1}}^{t_i} \int_{t_{j-1}}^{t_j} G(|t-s|)dsdt$$

DHAM Continues...

In our case, $G(t-s) = (t-s)^{-\gamma}$, we have

$$G_{ij} = \begin{cases} \frac{1}{(1-\gamma)(2-\gamma)}\left(\frac{T}{N}\right)^{2-\gamma}\{(i-j+1)^{2-\gamma} - 2(i-j)^{2-\gamma} + (i-j-1)^{2-\gamma}\} & i > j \\ \frac{1}{(1-\gamma)(2-\gamma)}\left(\frac{T}{N}\right)^{2-\gamma}\{(j-i+1)^{2-\gamma} - 2(j-i)^{2-\gamma} + (j-i-1)^{2-\gamma}\} & i < j \\ \frac{2}{(1-\gamma)(2-\gamma)}\left(\frac{T}{N}\right)^{2-\gamma} & i = j \end{cases}$$

Constraint becomes

$$\int_0^T v(t)dt = X \longrightarrow \sum_{i=1}^{N} v_i = \frac{NX}{T}$$

DHAM Continues...

The mth-order discrete homotopy derivatives at time t_i

$$v_i^m = v_i^{m-1} + \hbar \sum_{j=1}^{N} G_{ij} F_{ij}^{m-1}$$

where $F_{ij}^{m-1} = \frac{\partial^{m-1}}{\partial p^{m-1}} F\left(\sum_{k=0}^{\infty} v_j^k p^k, \sum_{l=0}^{\infty} v_i^l p^l \right) \Big|_{p=0}$.

nth-order approximated solution at time t_i

$$v_i^{(n)} = v_i^{(0)} + \sum_{m=1}^{n} v_i^m$$

DHAM Almost There...

Discrete expected cost

$$C(\Pi) \approx C(v^{(n)}) = \sum_{i=1}^{N}\sum_{j=1}^{N} v_i^{(n)} A_{ij} f\left(v_j^{(n)}\right) = v^{(n)} A f(v^{(n)})^\top$$

where

$$A_{ij} = \begin{cases} 0 & j > i \\ G_{ii}/2 & i = j \\ G_{ij} & j < i \end{cases}$$

Initial guess $v^{(0)}$ candidates

1. **VWAP:** $v_i^{(0)} = \frac{X}{TN}$
2. **GSS (linear solution)**

DHAM Results

Implementation Details

- Python 3.8.8[12]
- Symbolic computation of partials in SymPy[10]
- Dynamic programming for efficiency
- Obtained 6th order approximate solution

Quick result on DHAM implementation time.

DHAM Results

Implementation Details

- Python 3.8.8[12]
- Symbolic computation of partials in SymPy[10]
- Dynamic programming for efficiency
- Obtained 6th order approximate solution

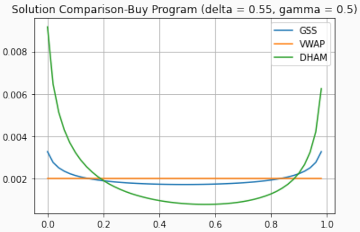
Quick result on DHAM implementation time.

Lemma (Computational Complexity of DHAM)

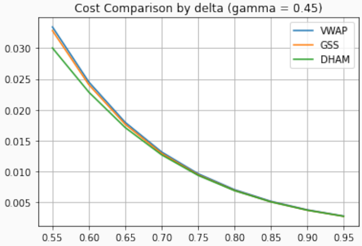
Let M be the pre-specified order of approximation. Using the polynomial dynamic programming approach, the DHAM solution is attainable in $\mathcal{O}(N^2 2^M)$ time.

Solution Comparison

We computed solutions on the (δ, γ) grid with $\delta \in \{0.55, 0.60, \cdots, 0.95\}$ and $\gamma \in \{0.45, 0.50, 0.55\}$ with $N = 50, X = 0.1, T = 1$.



(a) Solution shape comparison



(b) Solution expected cost comparison

Numerical Nonlinear Programming

Really a Nonlinear Program

$$\begin{array}{ll} \min & C(v) = vAf(v)^\top \\ \text{s.t.} & \sum_{i=1}^{N} v_i = \frac{XN}{T} \\ & v_i \in \mathbb{R} \end{array}$$

Implemented in Python[12].

Sequential Quadratic Programming

Really a Nonlinear Program

$$
\begin{array}{ll}
\min & C(v) = vAf(v)^\top \\
\text{s.t.} & \sum_{i=1}^{N} v_i = \frac{XN}{T} \\
 & v_i \in \mathbb{R}
\end{array}
$$

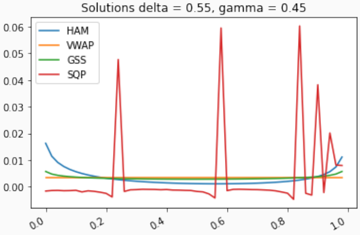
Implemented in Python[12].

Remarks

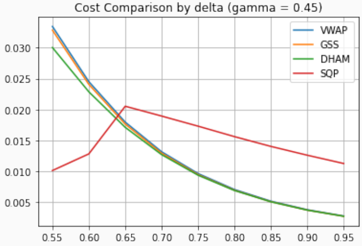
- When $0 < \gamma < 1$, not convex
- Aim to find local minima
- Constraints on v_i's sign will make it suffer convergence problem
- Iterative algorithm with an initial point

SQP Results

We computed SQP solutions on the same (δ, γ) grid with **GSS** solution as an initial guess.

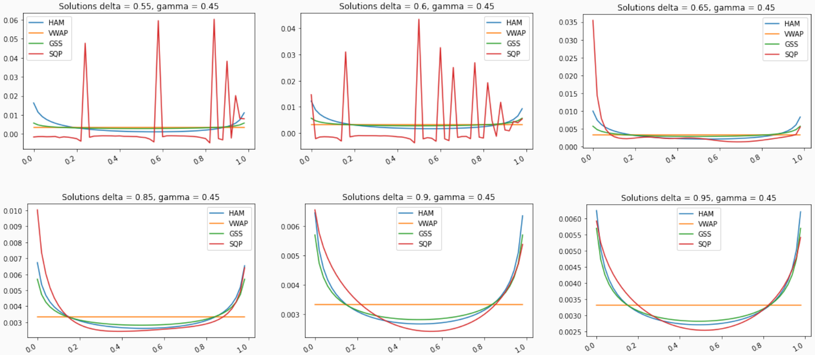


(a) Solution shape comparison



(b) Solution expected cost comparison

Convergence of Solutions



Transaction-triggered Price Manipulation

For a liquidation strategy v with $\int_0^T v(t)dt = X$, there exists t such that $v(t) < 0$.

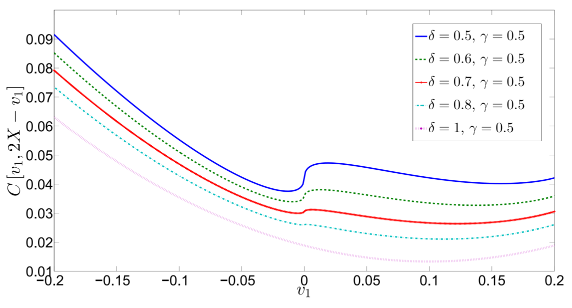


Figure: $N = 2$ example. SQP can find local minima by incuring $v_i < 0$ in a buy program near $\gamma + \delta = 1$[3]

Empirical Test

Market Data Calibration

Calibration of f

- Order data required (rare and expensive to find)
- Regression between ”impact” and ”volume”

Market Data Calibration

Calibration of f

- Order data required (rare and expensive to find)
- Regression between ”impact” and ”volume”

Propagator Model Calibration (Enzo 2012)[2]

Let $\mathbb{T}$ be an ordered partition of $[0, T]$ of size N. For $I_n \in \mathbb{T}, 1 \leq n \leq N$, the **aggregated volume** is

$$v_n = \sum_{i \in I_n} v_i^{tt}$$

the **normalized volume** is

$$v_n^* = \frac{\sum_{i \in I_n} v_i^{tt}}{\sum_{i \in I_n} |v_i^{tt}|}$$

Discrete Propagator Model

Recall that

$$S_j = S_0 + \sum_{k=0}^{j-1} \left(f(v_k^*) \cdot G(j-k) + \eta_k\right)$$

then,

$$r_j = S_{j+1} - S_j = \sum_{k=1}^{j-1} f(v_{j-k}^*)(G(k+1) - G(k)) + \eta_j$$

My Own Experiment

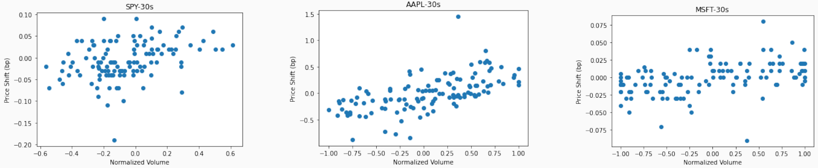


Figure: Scatter plots between r_j and v_j^* using 30-second interval among AAPL, MSFT, SPY on 2012-06-21 9:30AM-10:30AM EST. *Sample free data provided by NASDAQ Historical TotalView-ITCH*

Shape of f?

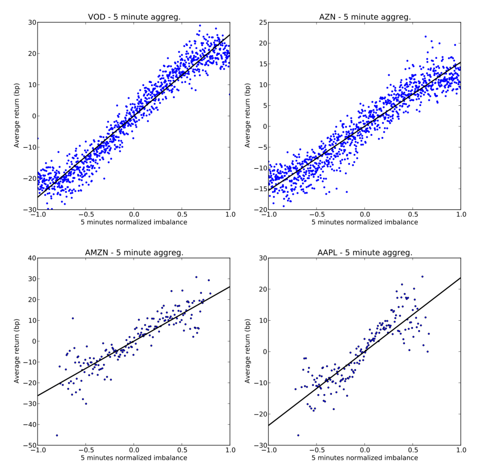


Figure: Similar tests on VOD, AZN, AMZN, AAPL with much larger data set (Enzo, Fabrizio 2012)[2]

TSLA Sunshine Event

CNBC

WATCH LIVE

TECH

Elon Musk sold about $6.9 billion in Tesla stock this week

PUBLISHED SAT, NOV 13 2021·9:11 AM EST UPDATED SAT, NOV 13 2021·10:00 AM EST

Todd Haselton
@ROBOTODD

WATCH LIVE

KEY POINTS

- Tesla CEO Elon Musk's trust sold about $1.2 billion in Tesla stock, according to financial filings posted Friday evening. He sold about $6.9 billion worth of stock in the company over the course of the week.
- Musk still holds more than 166 million shares in the company.
- Tesla stock declined 15.4% for the week, marking the company's worst one-week performance in 20 months.

TSLA Sunshine Event

Application of Execution Solutions

We get price series of TSLA from 11/08/2021 2:30PM to 11/12/2021 4:00PM. See how the price dynamic evolves based on our execution scheme versus what happened in real life. (Very hand-waving)

Simulated Result

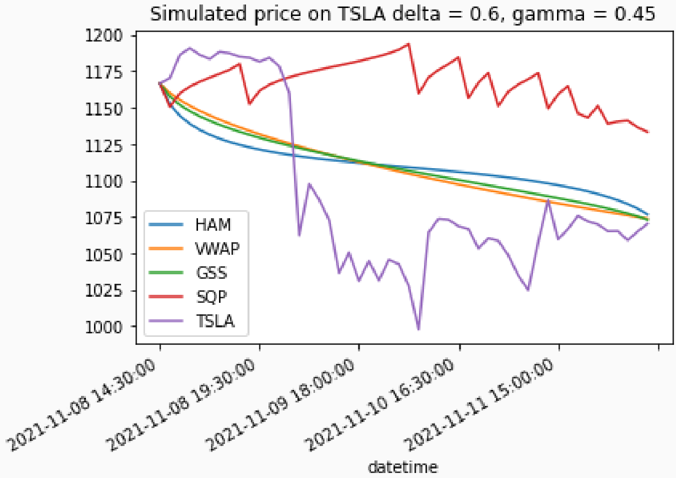


Figure: SQP solution seems suggesting the impossible

Possible Extensions

Non-exhaustive List

- Spread cost regularization (Lasso). See [3] Section 5.
- How to make HAM faster?
- How to have more rigorous test of execution scheme without goint directly to the market

Q&A Time!

Reference I

BOUCHAUD, J.-P., GEFEN, Y., POTTERS, M., AND WYART, M.
Fluctuations and response in financial markets: the subtle nature of "random" price changes.
Quantitative Finance 4, 2 (Apr 2004), 176–190.

BUSSETI, E., AND LILLO, F.
Calibration of optimal execution of financial transactions in the presence of transient market impact.
Journal of Statistical Mechanics: Theory and Experiment 2012, 09 (Sep 2012), P09010.

CURATO, G., GATHERAL, J., AND LILLO, F.
Optimal execution with nonlinear transient market impact, 2014.

Reference II

DANG, N.-M.
Optimal execution with transient impact.
Market Microstructure and Liquidity 03, 01 (2017), 1750008.

GATHERAL, J.
No-dynamic-arbitrage and market impact.
Quantitative Finance 10, 7 (2010), 749–759.

GATHERAL, J., AND SCHIED, A.
Dynamical models of market impact and algorithms for order execution.
SSRN Electronic Journal (01 2013).

Reference III

LIAO, S., AND SHERIF, S.
Beyond perturbation: Introduction to the homotopy analysis method.
Applied Mechanics Reviews 57, 5 (2004).

LIAO, S., AND SHERIF, S. A.
Beyond perturbation: Introduction to the homotopy analysis method.
Applied Mechanics Reviews 57 (09 2004), 25–.

LILLO, F., FARMER, J. D., AND MANTEGNA, R. N.
Master curve for price-impact function.
Nature 421, 6919 (Jan 2003), 129–130.

Reference IV

MEURER, A., SMITH, C. P., PAPROCKI, M., ČERTÍK, O., KIRPICHEV, S. B., ROCKLIN, M., KUMAR, A., IVANOV, S., MOORE, J. K., SINGH, S., RATHNAYAKE, T., VIG, S., GRANGER, B. E., MULLER, R. P., BONAZZI, F., GUPTA, H., VATS, S., JOHANSSON, F., PEDREGOSA, F., CURRY, M. J., TERREL, A. R., ROUČKA, V., SABOO, A., FERNANDO, I., KULAL, S., CIMRMAN, R., AND SCOPATZ, A.
Sympy: symbolic computing in python.
PeerJ Computer Science 3 (Jan. 2017), e103.

OBIZHAEVA, A. A., AND WANG, J.
Optimal trading strategy and supply/demand dynamics.
Journal of Financial Markets 16, 1 (2013), 1–32.

Reference V

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors.
SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.
Nature Methods 17 (2020), 261–272.