

Brownian Bridge Vectorial Simulation

J. Beleza Sousa, M. L. Esquível, R. M. Gaspar

February 28, 2012

Abstract

The iterative simulation of the Brownian bridge is well known. In this paper we present a vectorial simulation alternative. The vectorial alternative is based on Gaussian processes for machine learning regression and is suitable for interpreted programming languages implementations.

We extend the vectorial simulation to other Gaussian processes, namely, sequences of Brownian bridges, path dependent geometric Brownian motion and path dependent Ornstein-Uhlenbeck mean reversion process.

1 Introduction

Interpreted programming languages like Sage, Octave, Mathematica and Matlab are important frameworks in nowadays research and development.

In these programming languages it is crucial to use vectorial algorithms instead of iterative ones in order to achieve the execution speeds of compiled languages. This is because vectorial operations are typically supported by built-in functions which are implemented by optimized machine code.

The iterative simulation of the Brownian bridge is well known [3][1]. In this paper we present a vectorial simulation alternative. The vectorial alternative is based on Gaussian processes for machine learning regression and is suitable for interpreted programming languages implementations.

We extend the vectorial simulation to other Gaussian processes, namely, sequences of Brownian bridges, path dependent geometric Brownian motion and path dependent Ornstein-Uhlenbeck mean reversion process.

We illustrate the flexibility of the vectorial simulation by creating a 2D Wiener process representation of a Norbert Wiener photograph.

2 Brownian bridge iterative simulation

A Brownian bridge is a standard Brownian motion W conditioned to $W(1) = 0$. The Brownian bridge condition $W(1) = 0$ can be generalized to other time instants greater than zero and to other values besides zero.

The standard Brownian motion W , defined in \mathbb{R}_0^+ , is also called a Wiener process [2] and has the following properties:

1. $W(0) = 0$;
2. W has independent increments, i.e. if $r < s \leq t < u$ then $W(u) - W(t)$ and $W(s) - W(r)$ are independent random variables;
3. For $s < t$ the random variable $W(t) - W(s)$ has the Gaussian distribution $\mathcal{N}(0, \sqrt{t - s})$;
4. W has continuous trajectories.

In addition, the Wiener process is a Gaussian process with mean function $m(t)$ and covariance function $cov(s, t)$:

$$m(t) = 0, \tag{1}$$

$$cov(s, t) = \min(s, t). \tag{2}$$

In order to illustrate the iterative simulation of a Brownian bridge trajectory B , consider that at some step we have $0 < u < s < t < 1$, $B(u) = \alpha$, $B(t) = \beta$ and we want to simulate the value $B(s)$ [3]. Given the Wiener process properties, the random vector $[B(u)B(s)B(t)]^T$ is Gaussian with mean vector and covariance matrix:

$$\begin{bmatrix} B(u) \\ B(s) \\ B(t) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} u & u & u \\ u & s & s \\ u & s & t \end{bmatrix} \right) \tag{3}$$

Therefore the conditional distribution $B(s)|B(u), B(t)$ is given by:

$$B(s)|B(u), B(t) \sim \mathcal{N}\left(\frac{(t-s)\alpha + (s-u)\beta}{t-u}, \frac{(s-u)(t-s)}{t-u}\right) \quad (4)$$

Thus, the value $B(s)$ is simulated by:

$$B(s) = \frac{(t-s)\alpha + (s-u)\beta}{t-u} + \sqrt{\frac{(s-u)(t-s)}{t-u}}Z, \quad (5)$$

where $Z \sim \mathcal{N}(0, 1)$ is an increment independent of all Z values previously used in the simulation.

Finally, the iterative simulation of a Brownian bridge trajectory consists of starting with $u = 0$, $t = 1$, $B(0) = 0$, $B(1) = 0$ and iteratively filling a trajectory sample at time s (between u and t) with Equation 5, then moving one of the end points to the simulated sample and repeating the process until all samples trajectory are filled.

Figure 1 shows a sequence of 500 simulated independent Gaussian $\mathcal{N}(0, 1)$ increments (white noise), and the corresponding simulated Wiener process and Brownian bridge trajectories (sampled uniformly 500 times between zero and one). The Brownian bridge trajectory was simulated by the iterative procedure above.

3 Gaussian processes for machine learning

The goal of Gaussian processes for machine learning regression is to find the non linear unknown mapping $y = f(\mathbf{x})$, from data (\mathbf{X}, \mathbf{y}) , using Gaussian distributions over functions [4]:

$$\mathcal{GP} \sim \mathcal{N}(m(\mathbf{x}), cov(\mathbf{x}_1, \mathbf{x}_2)). \quad (6)$$

The pair (\mathbf{X}, \mathbf{y}) is the training set. The matrix \mathbf{X} collects a set of n vectors \mathbf{x} where the value $y = f(\mathbf{x})$ was observed. The corresponding y values are collected in vector \mathbf{y} .

The set of vectors \mathbf{x}^* where the values $y^* = f(\mathbf{x}^*)$ were not observed, is collected in matrix \mathbf{X}^* . The matrix \mathbf{X}^* is the test set.

The regression function is the mean function of the process defined by all the trajectories of the process of Equation 6 that passes through the training set. The regression confidence is the corresponding covariance function.

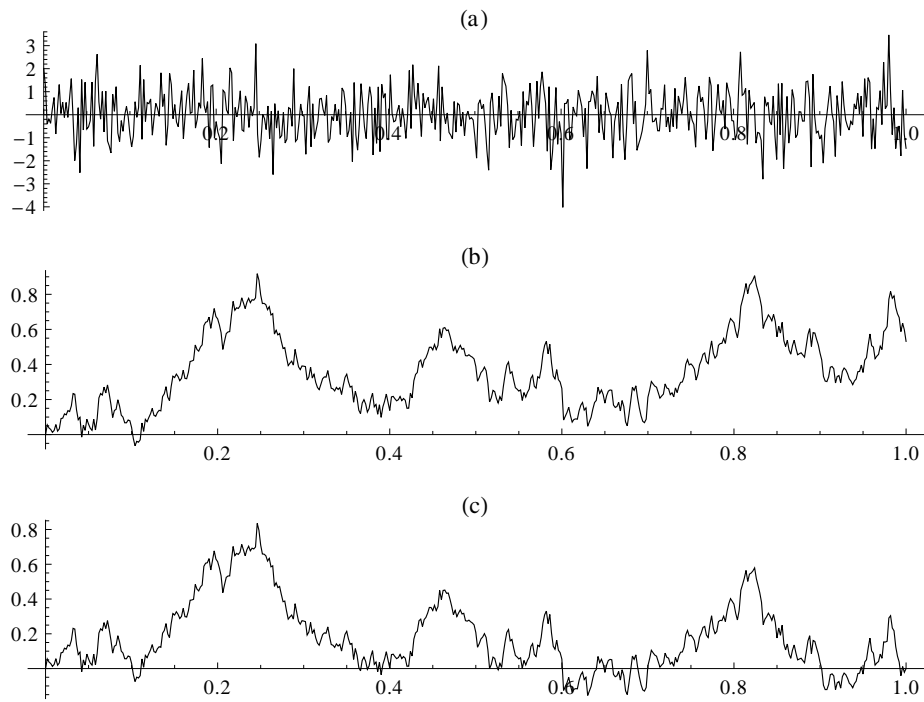


Figure 1: Simulated trajectories of: (a) white noise; (b) the corresponding Wiener process; (c) the corresponding Brownian bridge.

As presented in the previous section, the Wiener process is a scalar Gaussian process

$$W \sim \mathcal{N}(m(t), cov(s, t)) \quad (7)$$

where $m(t)$ is given by Equation 1 and $cov(s, t)$ by Equation 2.

In this case the mapping f is the scalar mapping $y = f(t)$, where y is the value of W at time t . This reduces the training set to the pair of vectors (\mathbf{t}, \mathbf{y}) , and the test set to vector \mathbf{t}^* .

Since the process is Gaussian [4]

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{m} \\ \mathbf{m}^* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{bmatrix} \right) \quad (8)$$

and

$$p(\mathbf{y}^* | \mathbf{t}^*, \mathbf{t}, \mathbf{y}) \sim \mathcal{N}(\mathbf{m}^* + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*) \quad (9)$$

where \mathbf{m} and \mathbf{m}^* are mean vectors of train and test sets, \mathbf{K} is the train set covariance matrix, \mathbf{K}_* the train-test covariance matrix and \mathbf{K}_{**} the test set covariance matrix.

The conditional distribution

$$p(\mathbf{y}^* | \mathbf{t}^*, \mathbf{t}, \mathbf{y}) \quad (10)$$

corresponds to the posterior process on the data

$$\mathcal{GP}_{\mathcal{D}} \sim \mathcal{N}(m_{\mathcal{D}}(t), cov_{\mathcal{D}}(s, t)) \quad (11)$$

where

$$m_{\mathcal{D}}(t) = m(t) + \mathbf{K}_{\mathbf{t},t}^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{m}) \quad (12)$$

and

$$cov_{\mathcal{D}}(s, t) = cov(s, t) - \mathbf{K}_{\mathbf{t},s}^T \mathbf{K}^{-1} \mathbf{K}_{\mathbf{t},t} \quad (13)$$

where $\mathbf{K}_{\mathbf{t},t}$ is a covariance vector between every training instant and t .

Equation 12 is the regression function while Equation 13 is the regression confidence. Equations 12 and 13 are the central equations of Gaussian processes for machine learning regression.

Figure 2 shows an example of Gaussian processes for machine learning regression with the Wiener process.

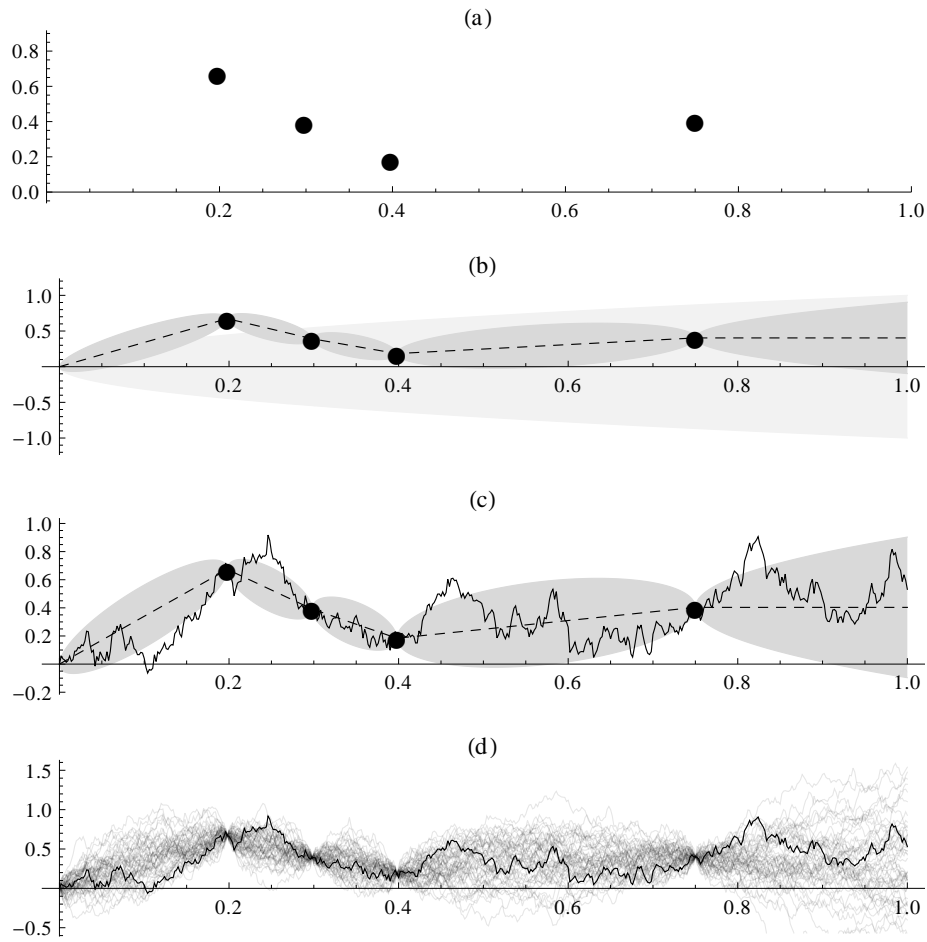


Figure 2: Gaussian processes for machine learning regression with the Wiener process: (a) training set; (b) regression function, two standard deviation confidence regression band (gray) and the not conditioned, mean centered, two standard deviation band (light gray); (c) training set trajectory; (d) simulated Wiener process trajectories passing through the training set.

4 Browning bridge vectorial simulation

Joining Sections 2 and 3 the vectorial simulation of a Brownian bridge trajectory is as follows.

Considering:

1. the Wiener process W with mean and covariance functions given by Equations 1 and 2;
2. the training set with the single pair $(t, y) = (1, 0)$ corresponding to the Brownian bridge condition $W(1) = 0$;
3. the test vector $\mathbf{t}^* = [t_1, t_2, \dots, t_n]^T$ where t_1, t_2, \dots, t_n are the time instants where to sample the Brownian bridge trajectory.

The Brownian bridge process B is Gaussian and the random vector $\mathbf{B} = B(\mathbf{t}^*)$ is also Gaussian

$$\mathbf{B} \sim \mathcal{N}(\mathbf{m}_{\mathcal{D}}, \mathbf{cov}_{\mathcal{D}}) \quad (14)$$

where $\mathbf{m}_{\mathcal{D}}$ is \mathbf{B} mean vector and $\mathbf{cov}_{\mathcal{D}}$ is \mathbf{B} covariance matrix. The element i of vector $\mathbf{m}_{\mathcal{D}}$ is given by

$$\mathbf{m}_{\mathcal{D}i} = m_{\mathcal{D}}(t_i) \quad (15)$$

and the element i, j of matrix $\mathbf{cov}_{\mathcal{D}}$ is given by

$$\mathbf{cov}_{\mathcal{D}i,j} = cov_{\mathcal{D}}(t_i, t_j). \quad (16)$$

Functions $m_{\mathcal{D}}(t_i)$ and $cov_{\mathcal{D}}(t_i, t_j)$ are those of Equations 1 and 2.

Therefore a Brownian bridge trajectory can be simulated as any other Gaussian vector [3], using:

$$\mathbf{B} = \mathbf{m}_{\mathcal{D}} + \mathbf{C}\mathbf{Z} \quad (17)$$

where \mathbf{C} is the Cholesky decomposition of $\mathbf{cov}_{\mathcal{D}}$ and \mathbf{Z} is a sample of the Gaussian random vector $\mathcal{N}(0, I)$.

Figure 3 shows a couple of Brownian bridge trajectories simulated with the vectorial Equation 17. The solid black one was simulated with the Gaussian increments of Figure 1. Since Equation 17 is just a vectorial alternative to the iterative procedure of section 2, the solid black trajectory is, as it would be expected, equal to the Brownian bridge trajectory of Figure 1.

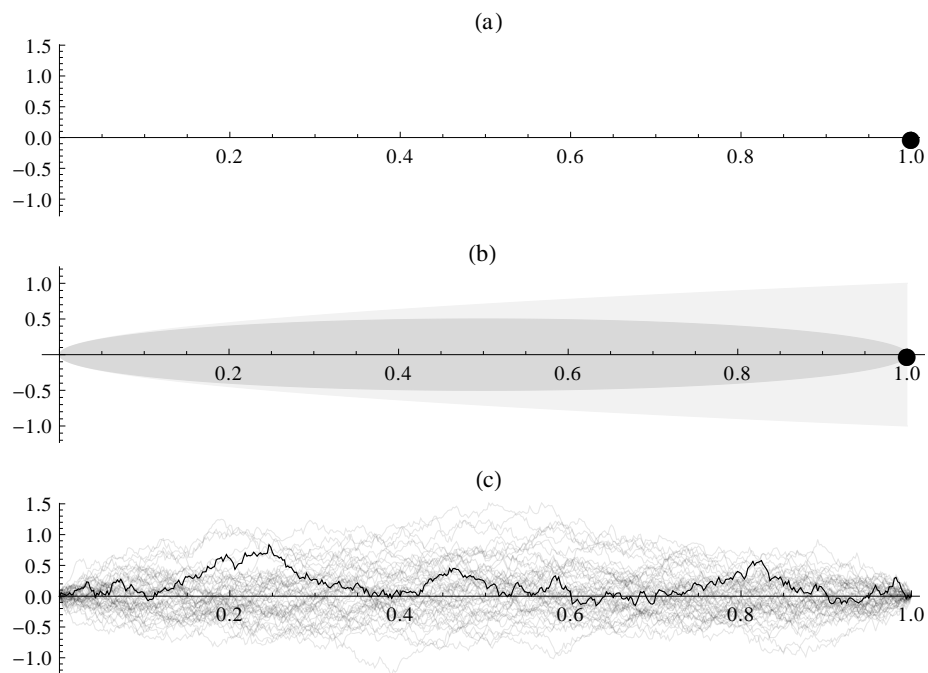


Figure 3: Brownian bridge trajectories simulated with the vectorial Equation 17: (a) training set; (b) regression function, two standard deviation confidence regression band (gray) and the not conditioned, mean centered, two standard deviation band (light gray); (c) Brownian bridge trajectories.

(a) System

CPU	Intel Core2 CPU 6300 1.86GHz
Memory	4GB
OS	Linux x86 (32bit)
Language	Mathematica 8.0.4.0

(b) Task

Simulation	Brownian bridge trajectories
Number of trajectories	1000
Number of samples per trajectory	1000 (uniformly)

(c) Execution Times (in seconds)

Iterative	32.31s
Vectorial	3.49s

Table 1: Iterative and vectorial execution times comparison.

5 Execution time comparison

In order to compare the execution times of the iterative Brownian bridge simulation procedure of Section 2 and the vectorial procedure of Section 4 under an interpreted language framework, we implemented both using the Mathematica 8 language [5] and tested the two alternatives on the task of generating 1000 Brownian bridges sampled uniformly 1000 times.

Table 1 describes: (a) the execution system; (b) the task; (c) the execution times of both alternatives.

As it would be expected, the Brownian bridge vectorial simulation with the Mathematica 8 language is faster than the iterative alternative. In the particular case tested, approximately 10 times faster. As mentioned before, this is because vectorial operations are supported by built-in functions which are implemented by optimized machine code.

6 Extensions

It is clear by the Brownian bridge vectorial construction that the simulation procedure can be naturally extended in the following 3 ways:

1. considering a condition different from $W(1) = 0$ (either in the time instant and its value);
2. considering more than one conditions (sequences of bridges);
3. considering other Gaussian processes besides the Wiener process (considering mean and covariance functions different from the Wiener process ones).

The first two ways were already illustrated by Figure 2(d) where there were a total of four conditions, different from the Brownian bridge condition.

Regarding the third way Figures 4 and 5 illustrate the same example of Figure 2, but now for geometric Brownian motion and Ornstein-Uhlenbeck mean reversion process. We choose this two processes for their importance in modelling stock prices and interest rates, respectively. The simulation procedure is the same of the example in Figure 2, except that the appropriate mean and covariance functions are used.

In the geometric Brownian motion case, the simulation was done for the underlying log normal process, which is a Gaussian process with mean and covariance functions given by:

$$m(t) = \left(\mu - \frac{\sigma^2}{2} \right) t \quad (18)$$

and

$$cov(s, t) = \sigma^2 \min(s, t) \quad (19)$$

where μ is the drift and σ the volatility. The geometric Brownian motion trajectories were obtained by taking the exponential of the log normal ones and multiplying by the process initial value x_0 . The values used in the simulation examples of Figure 4 were: $x_0 = 0.5$; $\mu = 1.0$ and $\sigma = 1.0$.

In the Ornstein-Uhlenbeck mean reversion process case, the mean and covariance functions are given by:

$$m(t) = x_0 e^{-kt} + \theta(1 - e^{-kt}) \quad (20)$$

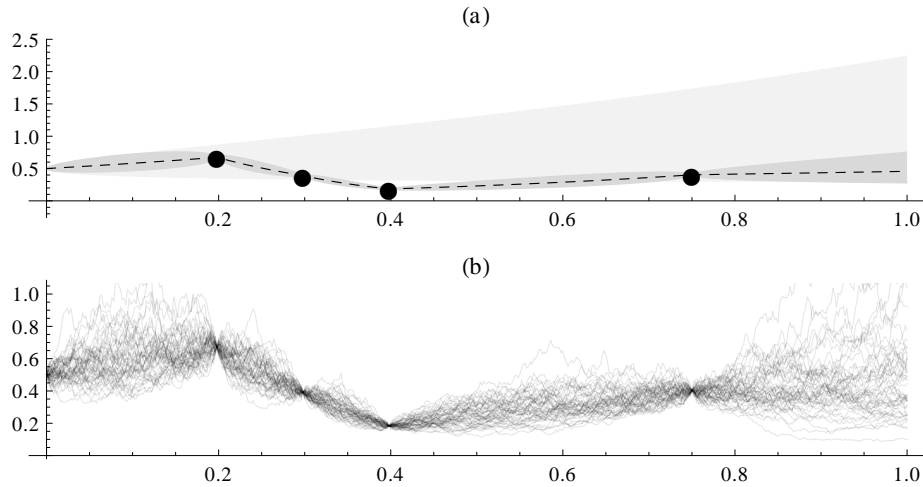


Figure 4: Gaussian processes for machine learning regression with the geometric Brownian motion: (a) training set, regression function, two standard deviation confidence regression band (gray) and the not conditioned, mean centered, two standard deviation band (light gray); (b) simulated trajectories passing through the training set.

and

$$cov(s, t) = \frac{\sigma^2}{2k} e^{-k(s+t)} (e^{2k \min(s,t)} - 1) \quad (21)$$

where x_0 is the process initial value, k is the mean reversion velocity, θ is the mean reversion level and σ the volatility. The values used in the simulation examples of Figure 5 were: $x_0 = 0.5$; $k = 2.0$; $\theta = 0.1$ and $\sigma = 0.2$.

7 Application

In this section we illustrate the great flexibility of the vectorial bridge trajectory simulation by constructing a 2D Wiener process single path representation of a Norbert Wiener photo. The steps taken to construct the representation are the following:

1. Choose a white background photo.

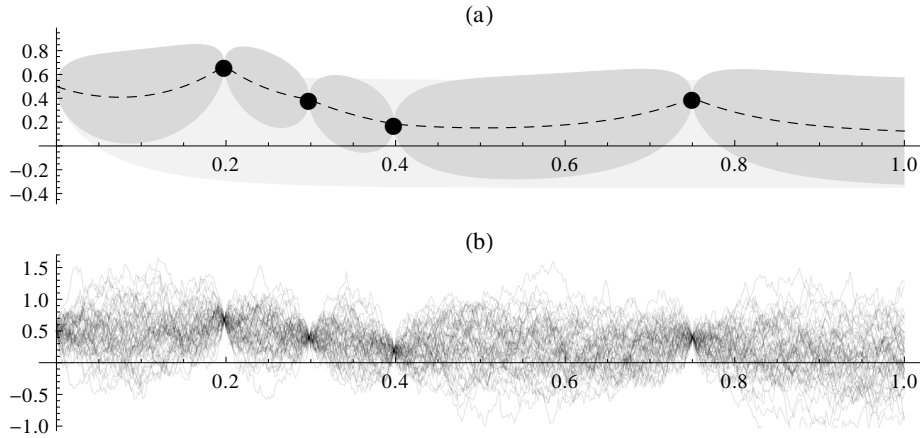


Figure 5: Gaussian processes for machine learning regression with the Ornstein-Uhlenbeck mean reversion process: (a) training set, regression function, two standard deviation confidence regression band (gray) and the not conditioned, mean centered, two standard deviation band (light gray); (b) simulated trajectories passing through the training set.

2. Obtain a binarized with dithering version of the photo.
3. Obtain a possible sequence of nearest black pixels: starting at a random black pixel find its nearest black pixel neighbor; repeat the procedure from the found neighbor, not considering the pixels already processed, until reaching the last unprocessed pixel.
4. Consider the black pixels coordinates, x and y , as the conditioning constraints.
5. For the sequence of the x coordinate constraints, simulate a Wiener process trajectory by sampling uniformly 50 times each successive pair of constraints.
6. Repeat the previous step for the y coordinate (using increments independent from those used for the x coordinate).
7. Plot the y coordinate trajectory as a function of the x coordinate trajectory.

Figure 6 shows the resulting image.



Figure 6: 2D Wiener process single path representation of a Norbert Wiener photo.

8 Conclusions

The contribution of the present paper is twofold:

1. it presents a vectorial alternative to the iterative simulation of Brownian bridge trajectories which is relevant regarding the execution speed of interpreted programming languages implementations;
2. it extends in a natural way the vectorial alternative to the class of path dependent Gaussian processes leading to a trajectories simulation vectorial framework of those processes.

References

- [1] Quantlib - a free/open-source library for quantitative finance, 2012.
- [2] Tomas Björk. *Arbitrage Theory in Continuous Time*. Oxford University Press, 2nd edition, 2004.
- [3] Paul Glasserman. *Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability) (v. 53)*. Springer, 2003.
- [4] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, 2005.
- [5] Inc. Wolfram Research. Mathematica edition: Version 8.0.4.0, 2011.