Model Predictive Control

Lecture 03A - Stochastic Differential Equations (SDEs)

John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

02619 Model Predictive Control

Deterministic and Stochastic Differential Equations

- Ordinary differential equations (ODEs)

$$dx(t) = f(x(t))dt$$

- Stochastic differential equations (SDEs)

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$

drift

= normal dist

- $\boldsymbol{\omega}(t)$ is a standard Wiener process

Deterministic and Stochastic Differential Equations

- Ordinary differential equations (ODEs)

$$dx(t) = f(x(t))dt$$

- Stochastic differential equations (SDEs) with state dependent diffusion

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$

- Stochastic differential equations (SDEs) with state independent diffusion

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + \sigma d\boldsymbol{\omega}(t)$$

  This is a special case, where $g(\boldsymbol{x}(t)) = \sigma$

Stochastic Differential Equation - Numerical Methods

- White noise
$$d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)$$
- Stochastic Differential Equation (SDE)
$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$
- Integral form
$$\boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k) = \overbrace{\int_{t_k}^{t_{k+1}} f(\boldsymbol{x}(t))dt}^{\text{Riemann integral}} + \overbrace{\int_{t_k}^{t_{k+1}} g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)}^{\text{Ito integral}}$$

  Riemann integral: The integrand can be evaluated for any t in $t_k \leq t \leq t_{k+1}$
  Ito integral: The integrand must be evaluated in $t = t_k$ (i.e. the LHS)
- Discretized white noise
$$\Delta\boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$
- Explicit-explicit method (Euler-Maruyama)
$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = f(\boldsymbol{x}_k)\Delta t_k + g(\boldsymbol{x}_k)\Delta\boldsymbol{w}_k$$
- Implicit-explicit method
$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = f(\boldsymbol{x}_{k+1})\Delta t_k + g(\boldsymbol{x}_k)\Delta\boldsymbol{w}_k$$

Brownian Motion

Discrete-time Brownian Motion

- Random walk = discrete Brownian motion
- Mathematical model
  - Initial value
    $$x_0 = 0$$
  - Random disturbance
    $$\boldsymbol{w}_k \sim F_{iid}(\cdot)$$
  - Difference equation
    $$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{w}_k$$
- Numerical procedure
  - $x_0 = 0$
  - for $k = 0, 1, 2, \ldots$
    Generate w_k using a random number generator
    $x_{k+1} = x_k + w_k$

Discrete-time Brownian Motion - Mean and Variance

- Difference equation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{w}_k, \quad \boldsymbol{x}_0 = x_0 = 0$$

- Equivalent expression

$$\boldsymbol{x}_k = \boldsymbol{w}_0 + \boldsymbol{w}_1 + \ldots + \boldsymbol{w}_{k-1} \qquad k \geq 1$$

- N_s different realizations of $\boldsymbol{x}_k$

$$x_k^i = w_0^i + w_1^i + \ldots + w_{k-1}^i \qquad k \geq 0 \quad i \in \{1, 2, \ldots, N_s\}$$

- Sample mean

$$\bar{x}_k = \frac{x_k^1 + x_k^2 + \ldots + x_k^{N_s}}{N_s}$$

- Sample variance

$$s_k^2 = \frac{(x_k^1 - \bar{x}_k)^2 + (x_k^2 - \bar{x}_k)^2 + \ldots + (x_k^{N_s} - \bar{x}_k)^2}{N_s - 1}$$

- Asymptotic limits: $\bar{x}_k \to 0$ and $s_k^2 \to k$ for $N_s \to \infty$

Scalar
Standard Brownian Motion = Standard Wiener Process

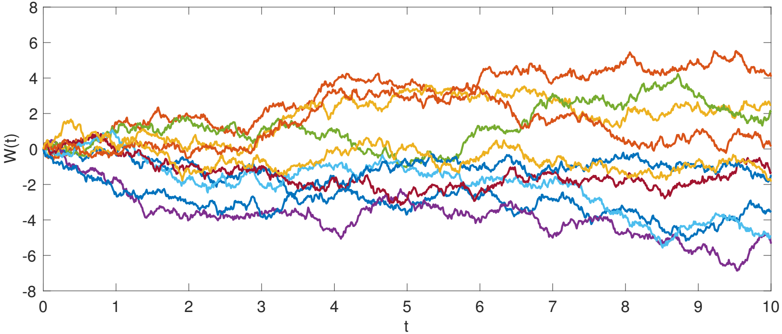
- Standard Brownian Motion = Standard Wiener Process
  A scalar standard Brownian motion, or standard Wiener process, over $[0,\, T]$ is a random variable $\boldsymbol{\omega}(t)$ that depends continuously on $t \in [0,\, T]$ and satisfies the following conditions
  - $\boldsymbol{\omega}(0) = 0$ (with probability 1)
  - $0 \leq s < t \leq T : [\boldsymbol{\omega}(t) - \boldsymbol{\omega}(s)] \sim N(0, t - s)$
  - For $0 \leq s < t < u < v \leq T$, the increments $\boldsymbol{\omega}(t) - \boldsymbol{\omega}(s)$ and $\boldsymbol{\omega}(v) - \boldsymbol{\omega}(u)$ are independent
- White noise
  Let $\boldsymbol{\omega}(t)$ be scalar standard Brownian motion and let

$$\boldsymbol{\omega}(t) = \int_0^t d\boldsymbol{\omega}(s)$$

  then $d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)$ is white noise
- Brownian motion is integrated white noise

Realizations of Standard Brownian Motion

```
% Scalar Standard Brownian Motion = Standard Wiener Process
Ns   = 10;        % Number of realizations
T    = 10;        % Final time
N    = 1000;      % Number of time steps
seed = 100;       % Seed for reproducibility

% Realization of Ns Standard Brownian Motions
rng(seed);
dt = T/N;
dW = sqrt(dt)*randn(Ns,N);
W  = cumsum(dW,2);
```



Scalar Standard Wiener Process

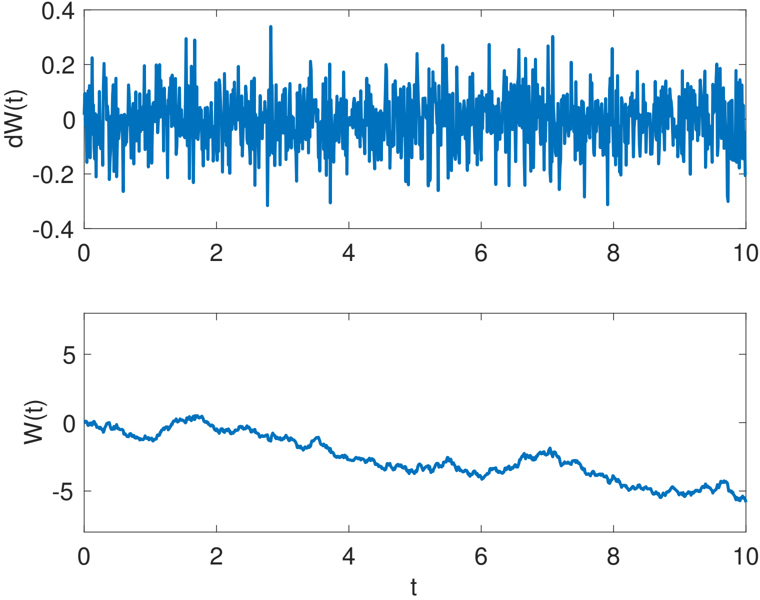
- N_s realizations of a scalar standard Wiener process

```
function [W,Tw,dW] = ScalarStdWienerProcess(T,N,Ns,seed)
% ScalarStdWienerProcess Ns realizations of a scalar std Wiener process
%
% Syntax: [W,Tw,dW] = ScalarStdWienerProcess(T,N,Ns,seed)
%         W    : Standard Wiener process in [0,T]
%         Tw   : Time points
%         dW   : White noise used to generate the Wiener process
%
%         T    : Final time
%         N    : Number of intervals
%         Ns   : Number of realizations
%         seed : To set the random number generator (optional)

if nargin == 4
    rng(seed);
end
dt = T/N;
dW = sqrt(dt)*randn(Ns,N);
W  = [zeros(Ns,1) cumsum(dW,2)];
Tw = 0:dt:T;
```

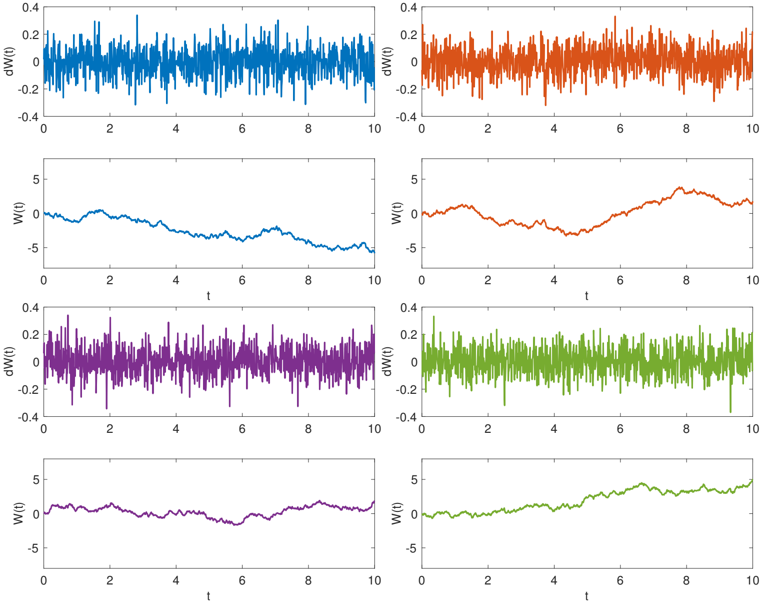
- Usage

```
[W,Tw,dW] = ScalarStdWienerProcess(10,1000,10,100);
plot(Tw,W,'linewidth',2);
```

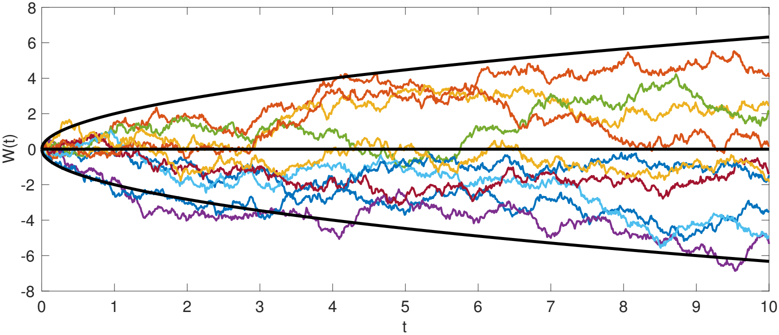
White Noise and Standard Brownian Motion



White Noise and Standard Brownian Motion



Realizations of Standard Brownian Motion



- Mean

$$\bar{\omega}(t) = E\boldsymbol{\omega}(t) = 0$$

- Variance

$$\sigma(t)^2 = V\boldsymbol{\omega}(t) = t$$

- Standard deviation

$$\sigma(t) = \sqrt{V\boldsymbol{\omega}(t)} = \sqrt{t}$$

- Plot: $\bar{\omega}(t) \pm 2\sigma(t)$

Scalar Sample Mean and Sample Variance (Std Deviation)

- Given $\{x_k^i\}_{i=1}^{N_s}$

$$\text{Sample mean:} \quad \bar{x}_k = \frac{1}{N_s} \sum_{i=1}^{N_s} x_k^i$$

$$\text{Sample std dev:} \quad s_k = \left(\frac{1}{N_s - 1} \sum_{i=1}^{N_s} (x_k^i - \bar{x}_k)^2 \right)^{1/2}$$

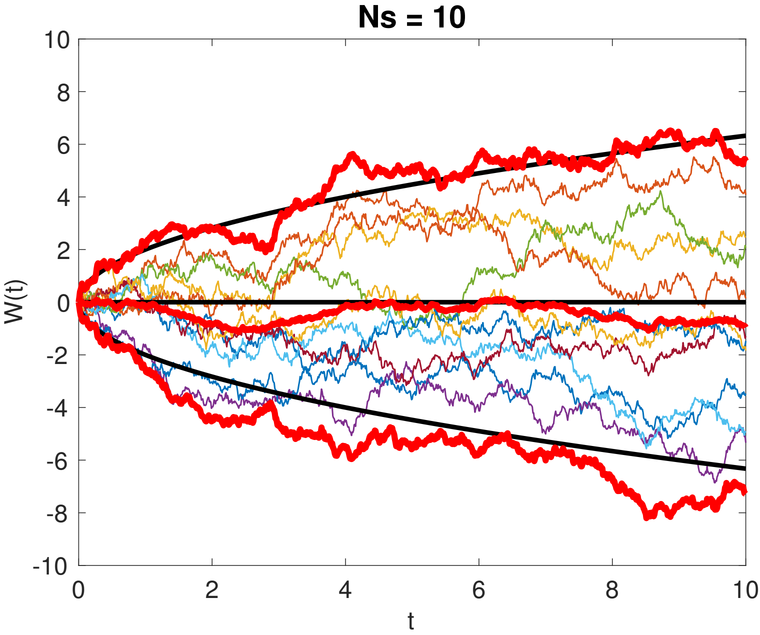
- Sample mean and standard deviation

```
function [xmean,s,xmeanp2s,xmeanm2s]=ScalarSampleMeanStdVar(x)

xmean    = mean(x,1);
s        = std(x,0,1);
xmeanp2s = xmean + 2*s;
xmeanm2s = xmean - 2*s;
```

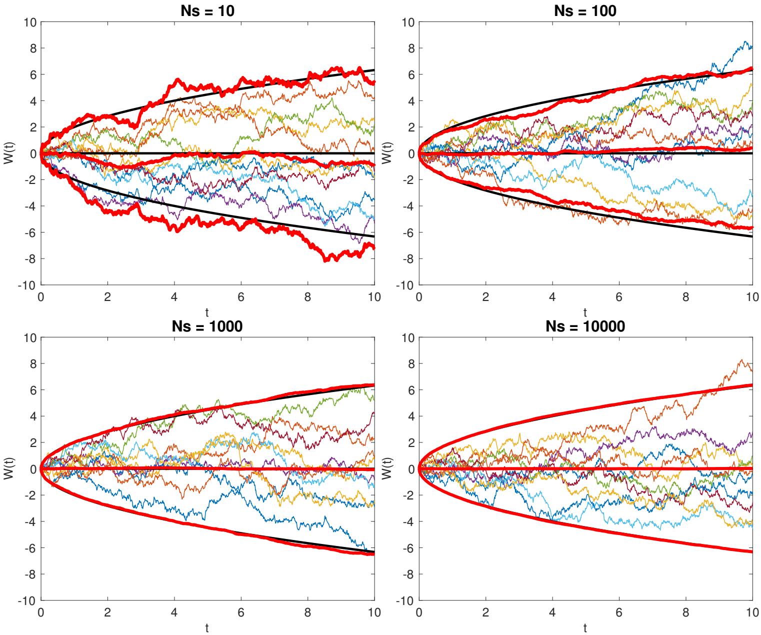
- Usage

```
[W,Tw,dW] = ScalarStdWienerProcess(T,N,Ns,seed);
[Wmean,sW,Wmeanp2sW,Wmeanm2sW]=ScalarSampleMeanStdVar(W);
```

Scalar Sample Mean and Sample Variance (Std Deviation)



Scalar Sample Mean and Sample Variance (Std Deviation)



Stochastic Differential Equations

Integrals

- Time axis

$$a = t_0 < t_1 < t_2 < \ldots < t_{N-1} < t_N = b$$

- Riemann integral (evaluated in the LHS)

$$\int_a^b f(t)dt = \lim_{\Delta t \to 0} \sum_{i=0}^{N-1} f(t_i)\Delta t_i$$

- Ito integral

$$\int_a^b f(t)d\boldsymbol{\omega}(t) = \lim_{\Delta t \to 0} \sum_{i=0}^{N-1} f(t_i)\Delta \boldsymbol{w}_i, \quad \Delta \boldsymbol{w}_i \sim N_{iid}(0, \Delta t_i)$$

- Brownian motion (a standard Wiener process)

$$\boldsymbol{\omega}(t) = \int_0^t d\boldsymbol{\omega}(s)$$

$d\boldsymbol{\omega}(t)$ is called white noise

Ito's Formula

- Ito's formula
  Let $y = f(t, x)$ then

$$dy = \frac{\partial f}{\partial t}(t, x)dt + \frac{\partial f}{\partial x}(t, x)dx + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t, x)dxdx$$

- The equivalence of the chain rule
- Reason that analytical solutions looks different than you are used to

Stochastic Differential Equation - Example 1

- Constant coefficient scalar differential equation

$$d\boldsymbol{x}(t) = \lambda dt + \sigma d\boldsymbol{\omega}(t), \quad \boldsymbol{x}(0) = 0$$

- Analytical solution

$$\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}(0) = \lambda t + \sigma\boldsymbol{\omega}(t)$$

- Integral equation

$$\boldsymbol{x}(t) - \boldsymbol{x}(0) = \overbrace{\int_0^t \lambda ds}^{\text{Riemann integral}} + \overbrace{\int_0^t \sigma d\boldsymbol{\omega}(s)}^{\text{Ito integral}}$$

- The solution is a combination of drift (λt) and the diffusion of the Brownian motion ($\sigma\boldsymbol{\omega}(t)$)

Stochastic Differential Equation - Example 1

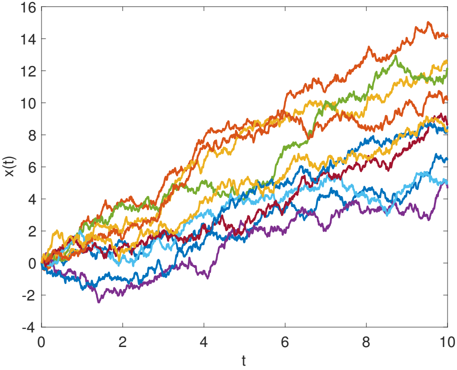
- Constant coefficient scalar differential equation

$$d\boldsymbol{x}(t) = \lambda dt + \sigma d\boldsymbol{\omega}(t), \quad \boldsymbol{x}(0) = 0$$

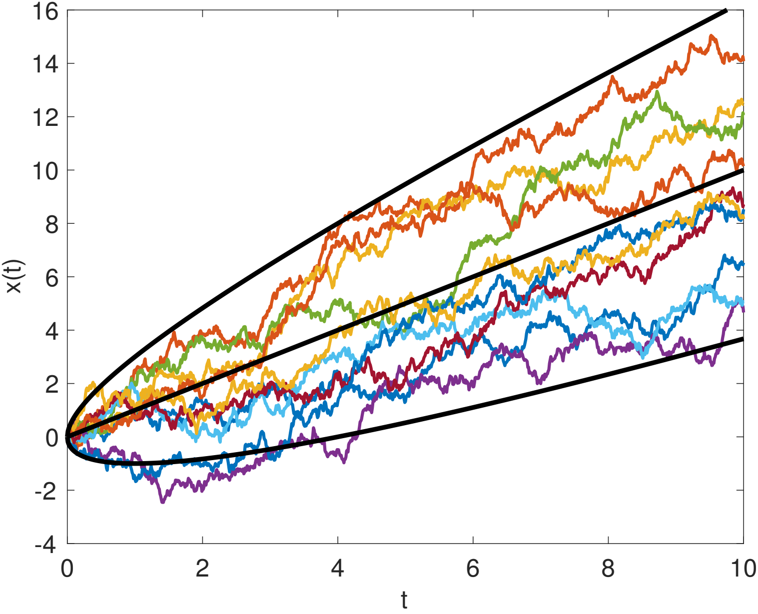
- Analytical solution

$$\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}(0) = \lambda t + \sigma \boldsymbol{\omega}(t)$$

- Plot of 10 realizations of the solution ($\lambda = 1$ and $\sigma = 1$)



Stochastic Differential Equation - Example 1



Stochastic Differential Equation - Example 2

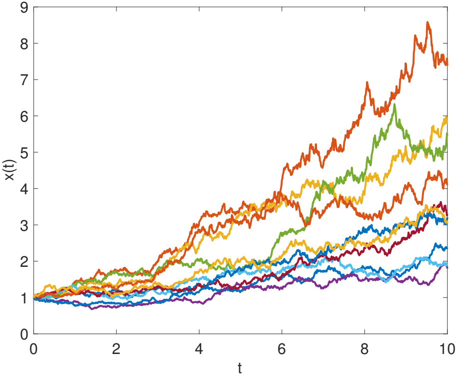
- Geometric Brownian motion

$$d\boldsymbol{x}(t) = \lambda \boldsymbol{x}(t)dt + \sigma \boldsymbol{x}(t)d\boldsymbol{\omega}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

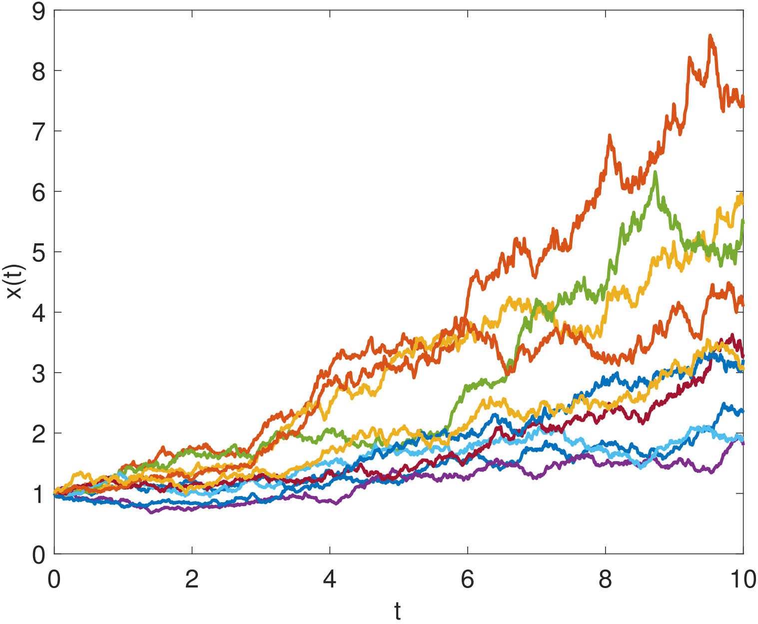
- Analytical solution

$$\boldsymbol{x}(t) = \exp\left(\left(\lambda - \frac{1}{2}\sigma^2\right)t + \sigma\boldsymbol{\omega}(t)\right)\boldsymbol{x}_0$$

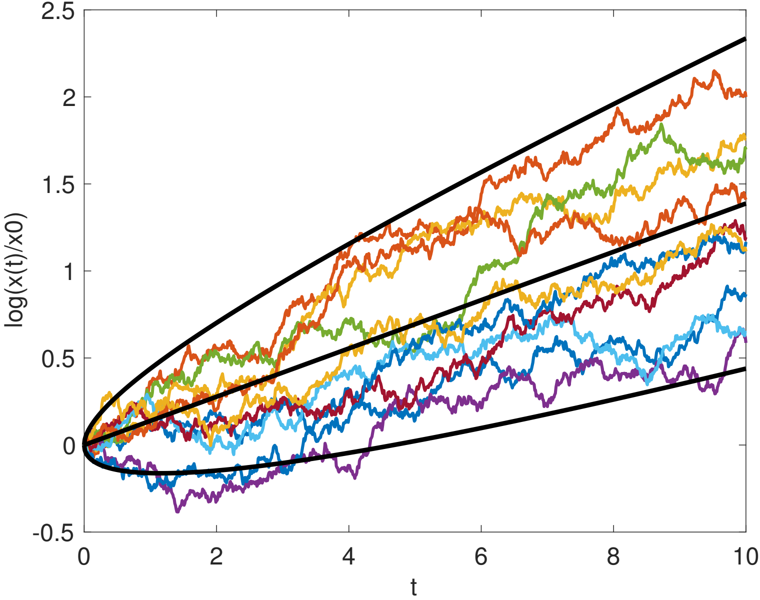
- Plot of 10 realizations of the solution ($\lambda = 0.15, \sigma = 0.15, x_0 = 1$)



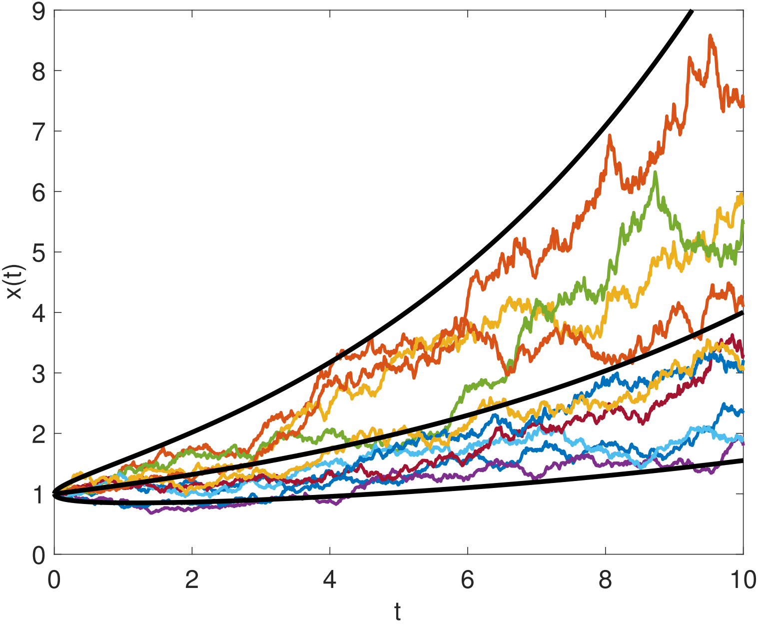
Stochastic Differential Equation - Example 2



Stochastic Differential Equation - Example 2



Stochastic Differential Equation - Example 2



Stochastic Differential Equation - Example 3

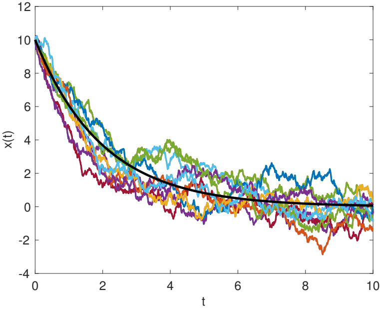
- Langevin equation

$$d\boldsymbol{x}(t) = \lambda \boldsymbol{x}(t)dt + \sigma d\boldsymbol{\omega}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

- No analytical solution exists
- The numerical solution (by the Euler-Maruyama method)

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = (\lambda \boldsymbol{x}_k)\,\Delta t_k + \sigma \Delta \boldsymbol{w}_k, \qquad \Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$

  is called the Ornstein-Uhlenbeck process
- Plot of 10 realizations of the solution ($\lambda = -0.5$,$\sigma = 1$,$x_0 = 10$)



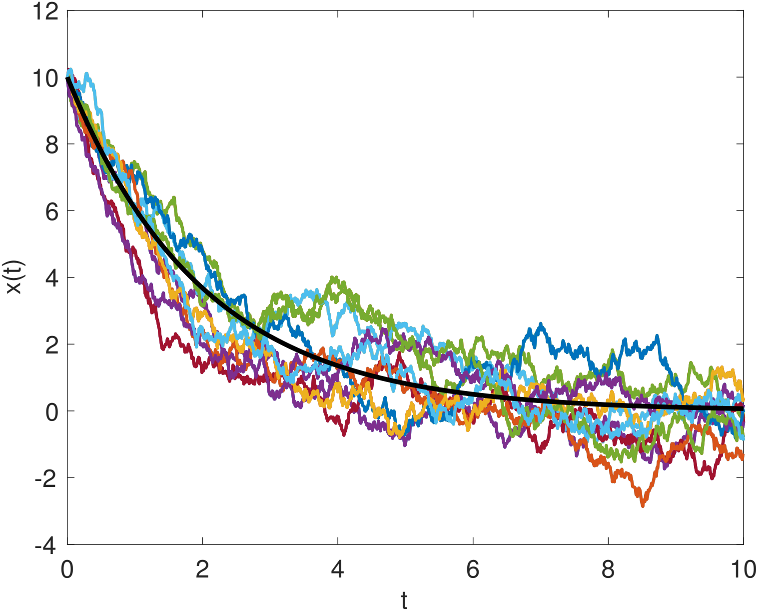
Matlab code

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + (\lambda \boldsymbol{x}_k)\,\Delta t_k + \sigma \Delta \boldsymbol{w}_k, \qquad \Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$

```
T    = 10;
N    = 1000;
Ns   = 10;
seed = 100;

x0     = 10;
lambda = -0.5;
sigma  = 1.0;

[W,Tw,dW]=ScalarStdWienerProcess(T,N,Ns,seed);

X = zeros(size(W));
for i=1:Ns;
   X(i,1) = x0;
   for k=1:N
       dt = Tw(k+1)-Tw(k);
       X(i,k+1) = X(i,k) + lambda*X(i,k)*dt + sigma*dW(i,k);
   end
end
```

Stochastic Differential Equation - Example 3

$\lambda = -0.5$, $\sigma = 1$, $x_0 = 10$



Order of a numerical method for SDEs

- An SDE solver has **order** p if the expected value of the error is of pth order in the time step size.
- $\forall t_N > 0$: $E\{|\boldsymbol{x}(t_N) - \boldsymbol{x}_N|\} = \mathcal{O}((\Delta t)^p)$ for $\Delta t \to 0$
  Analytical solution: $\boldsymbol{x}(t_N)$
  Numerical solution: $\boldsymbol{x}_N$
- Note that this is a concept related to the global error for ODE solvers

Euler-Maruyama Method (explicit-explicit)

- SDE

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$

- Numerical method (Euler-Maruyama)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f(\boldsymbol{x}_k)\Delta t_k + g(\boldsymbol{x}_k)\Delta \boldsymbol{w}_k$$

- Random number generation

$$\Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$

- Order: $p = \frac{1}{2}$

Matlab Implementation - SDE solver

```
function X = SDEeulerExplicitExplicit(ffun,gfun,T,x0,W,varargin)

N  = size(T,2)-1;
nx = size(x0,1);
X  = zeros(nx,N+1);

X(:,1) = x0;
for k=1:N
   dt = T(k+1)-T(k);
   dW = W(:,k+1)-W(:,k)
   f = feval(ffun,T(k),X(:,k),varargin{:});
   g = feval(gfun,T(k),X(:,k),varargin{:});
   X(:,k+1) = X(:,k) + f*dt + g*dW;
end
```

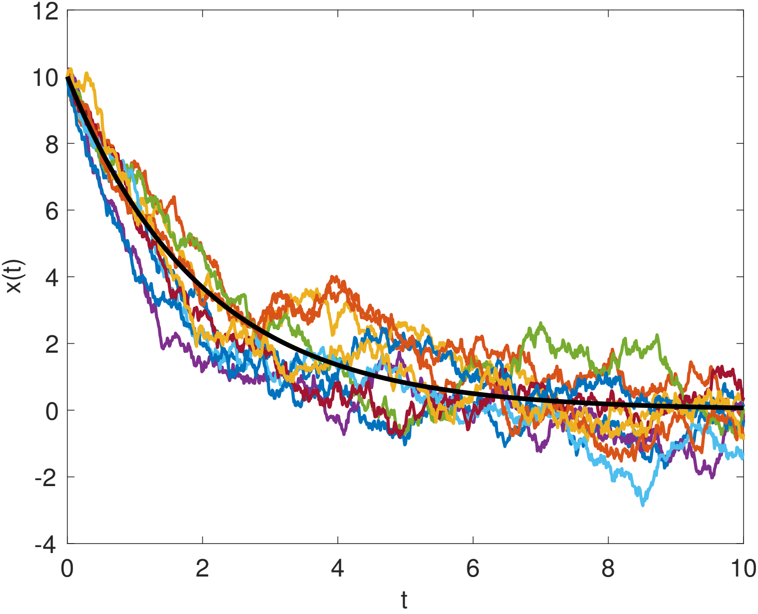
Matlab Implementation - Langevin - Driver, drift, diffusion

```matlab
T    = 10;
N    = 1000;
Ns   = 10;
seed = 100;

x0     = 10;
lambda = -0.5;
sigma  = 1.0;
p      = [lambda; sigma];

[W,Tw,dW]=ScalarStdWienerProcess(T,N,Ns,seed);
X = zeros(size(W));
for i=1:Ns
    X(i,:) = SDEeulerExplicitExplicit(...
                     @LangevinDrift,@LangevinDiffusion,...
                     Tw,x0,W(i,:),p);
end
```

```matlab
function f = LangevinDrift(t,x,p)

lambda = p(1);
f = lambda*x;
```

```matlab
function g = LangevinDiffusion(t,x,p)

sigma = p(2);
g = sigma;
```

Numerical Solution - Explicit-Explicit Method



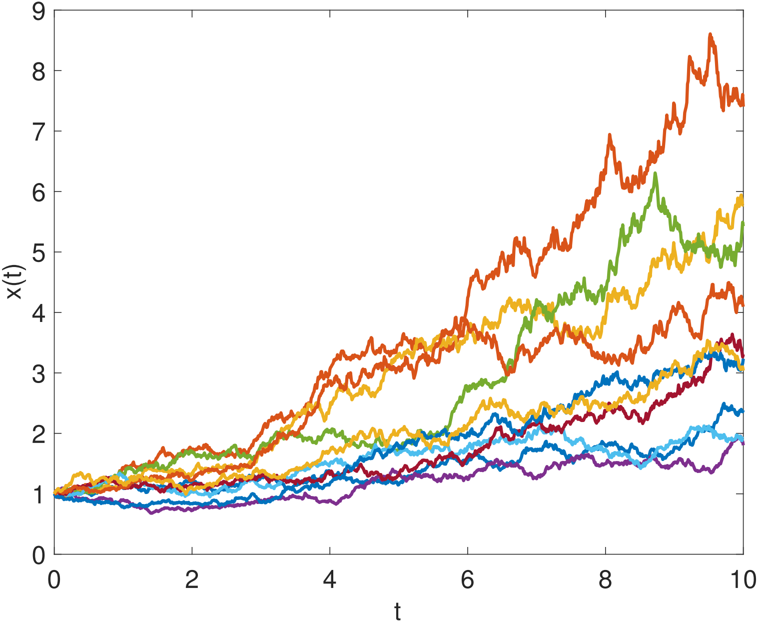
Matlab Implementation - Geometric Brownian Motion

```
T    = 10;
N    = 1000;
Ns   = 10;
seed = 100;

x0 = 1;
lambda = 0.15;
sigma  = 0.15;
p = [lambda; sigma];

[W,Tw,dW]=ScalarStdWienerProcess(T,N,Ns,seed);
X = zeros(size(W));
for i=1:Ns
   X(i,:) = SDEeulerExplicitExplicit(...
                  @GeometricBrownianDrift,@GeometricBrownianDiffusion,...
                  Tw,x0,W(i,:),p);
end
```

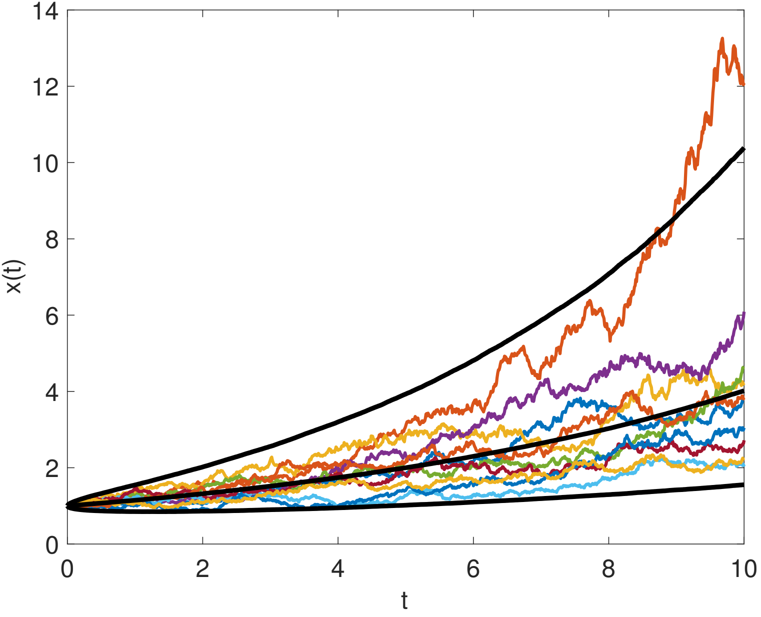
```
function f = ...
    GeometricBrownianDrift(t,x,p)

lambda = p(1);
f = lambda*x;
```

```
function g = ...
    GeometricBrownianDiffusion(t,x,p)

sigma = p(2);
g = sigma*x;
```

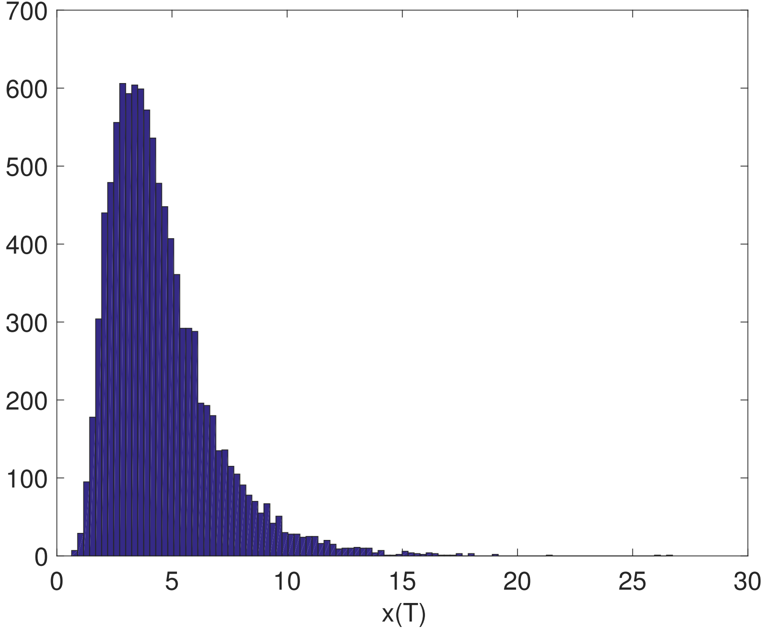
Numerical Solution - Explicit-Explicit Method



Numerical Solution - Explicit-Explicit Method



Numerical Solution - Distribution of final state, $x(T)$



Implicit-explicit Method

- SDE

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$

- Numerical method

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f(\boldsymbol{x}_{k+1})\Delta t_k + g(\boldsymbol{x}_k)\Delta \boldsymbol{w}_k$$

- Random number generation

$$\Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$

- Order: $p = \frac{1}{2}$

Newton Method in Implicit-Explicit SDE Solver

```
function [x,f,J] = SDENewtonSolver(ffun,t,dt,psi,xinit,tol,maxit,varargin)

I = eye(length(xinit));
x = xinit;
[f,J] = feval(ffun,t,x,varargin{:});
R = x - f*dt - psi;
it = 1;
while ( (norm(R,'inf') > tol) & (it <= maxit) )
    dRdx = I - J*dt;
    mdx = dRdx\R;
    x = x - mdx;
    [f,J] = feval(ffun,t,x,varargin{:});
    R = x - f*dt - psi;
    it = it+1;
end
```

Matlab Implementation

```matlab
function X = SDEeulerImplicitExplicit(ffun,gfun,T,x0,dW,varargin)

tol = 1.0e-8;
maxit = 10;

N  = size(T,2)-1;
nx = size(x0,1);
X  = zeros(nx,N+1);

X(:,1) = x0;
for k=1:N
    dt = T(k+1)-T(k);
    f = feval(ffun,T(k),X(:,k),varargin{:});
    g = feval(gfun,T(k),X(:,k),varargin{:});
    psi = X(:,k) + g*dW(:,k);
    xinit = psi + f*dt;
    [X(:,k+1),f,~] = SDENewtonSolver(...
                            ffun,...
                            T(:,k+1),dt,psi,xinit,...
                            tol,maxit,...
                            varargin{:});
end
```

Milstein Method (explicit)

- SDE

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$

- Numerical method (Milstein)

$$\begin{aligned}\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f(\boldsymbol{x}_k)\Delta t_k + g(\boldsymbol{x}_k)\Delta \boldsymbol{w}_k \\ + \frac{1}{2} g(\boldsymbol{x}_k)\frac{\partial g}{\partial x}(\boldsymbol{x}_k)((\Delta \boldsymbol{w}_k)^2 - \Delta t_k)\end{aligned}$$

- Random number generation

$$\Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$

- Order: $p = 1$

First-order Stochastic Runge-Kutta Method (explicit)

- SDE

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t)$$

- Numerical method (First-order stochastic Runge-Kutta)

$$\begin{aligned}
\boldsymbol{k}_k &= \boldsymbol{x}_k + g(\boldsymbol{x}_k)\sqrt{\Delta t_k} \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + f(\boldsymbol{x}_k)\Delta t_k + g(\boldsymbol{x}_k)\Delta \boldsymbol{w}_k \\
&\qquad\qquad + \frac{1}{2\sqrt{\Delta t_k}}\left[g(\boldsymbol{k}_k) - g(\boldsymbol{x}_k)\right]\left[(\Delta \boldsymbol{w}_k)^2 - \Delta t_k\right]
\end{aligned}$$

- Random number generation

$$\Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t_k)$$

- Order: $p = 1$

Summary

- Brownian motion (standard Wiener process)
- Riemann and Ito integrals
- Stochastic Differential Equation (SDE)
- Euler Explicit-Explicit Method
- Euler Implicit-Explicit Method
- Order