

## 1. 蒙特卡洛方法收敛性

对于  $f: \mathcal{R}^N \rightarrow \mathcal{R}$ , 定义  $\text{Var}_\rho[f] = \mathbb{E}_\rho[(f - \mathbb{E}_\rho[f])^2]$ , 我们有

$$\mathbb{E}\left[\frac{1}{J} \sum_{j=1}^J f(\theta^j) - \mathbb{E}[f]\right] = 0$$
$$\mathbb{E}\left[\left(\frac{1}{J} \sum_{j=1}^J f(\theta^j) - \mathbb{E}[f]\right)^2\right] = \frac{\text{Var}_\rho[f]}{J}$$

## 证明

由于  $\theta^j \sim \rho$ , 我们有

$$\mathbb{E}\left[\frac{1}{J} \sum_{j=1}^J f(\theta^j) - \mathbb{E}[f]\right] = \frac{1}{J} \sum_{j=1}^J \mathbb{E}[f] - \mathbb{E}[f] = 0$$

对于方差, 我们有

$$\mathbb{E}\left[\left(\frac{1}{J} \sum_{j=1}^J f(\theta^j) - \mathbb{E}[f]\right)^2\right] = \mathbb{E}\left[\frac{1}{J^2} \sum_{j=1}^J (f(\theta^j) - \mathbb{E}[f])^2\right]$$
$$= \frac{1}{J} \mathbb{E}\left[(f(\theta) - \mathbb{E}[f])^2\right]$$
$$= \frac{\text{Var}_\rho[f]}{J}$$

## 2. 重要性采样方法收敛性

定义  $L(\theta) = e^{-\Phi(\theta)}$ ,  $\rho^* = \frac{1}{Z} L(\theta) \rho(\theta)$ , 我们将证明

$$\sup_{|f_\infty| \leq 1} \mathbb{E}_\rho \left[ \left| \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] \right| \right] \leq 2 \frac{1 + \chi^2 \rho^* \|\rho\|}{J}$$
$$\sup_{|f_\infty| \leq 1} \mathbb{E}_\rho \left[ \left( \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] \right)^2 \right] \leq \frac{4 + \chi^2 \rho^* \|\rho\|}{J}$$

## 证明

我们有

$$\rho_{j|j}^*(f) = \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)}$$
$$\chi^2 \rho^* \|\rho\| = \frac{\rho^*}{\rho} - 1 = \frac{\mathbb{E}_\rho[L(\theta)^2]}{\mathbb{E}_\rho[L(\theta)]} - 1$$
$$\frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] = \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right) + \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z} f(\theta^j) - \mathbb{E}_{\rho^*}[f]$$

我们用  $\rho_{j|j}^*(f) \leq |f|_\infty$  和  $\mathbb{E}_\rho \left[ \left( \frac{1}{J} \sum_{j=1}^J f(\theta^j) - \mathbb{E}[f] \right)^2 \right] \leq \mathbb{E}_\rho[f(\theta^2)]$ , 对于方差

$$\mathbb{E}_\rho \left[ \left( \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] \right)^2 \right] = \mathbb{E}_\rho \left[ \left( \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right) + \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z} f(\theta^j) - \mathbb{E}_{\rho^*}[f] \right)^2 \right]$$
$$\leq 2 \mathbb{E}_\rho \left[ \left( \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} \right)^2 \right] \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right)^2 + \left(\frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z} f(\theta^j) - \mathbb{E}_{\rho^*}[f]\right)^2$$
$$\leq 2 \mathbb{E}_\rho \left[ \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right)^2 \right] + \left(\frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z} f(\theta^j) - \mathbb{E}_{\rho^*}[f]\right)^2$$
$$\leq \frac{2}{J} \mathbb{E}_\rho \left[ \frac{L(\theta)^2}{Z^2} \right] + \mathbb{E}_\rho \left[ \frac{L(\theta)^2 f(\theta)^2}{Z^2} \right]$$
$$\leq 4 \frac{\chi^2 \rho^* \|\rho\| + 1}{J}$$

对于偏差

$$\mathbb{E}_\rho \left[ \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] \right] = \mathbb{E}_\rho \left[ \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right) + \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z} f(\theta^j) - \mathbb{E}_{\rho^*}[f] \right]$$
$$= \mathbb{E}_\rho \left[ \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right) \right]$$
$$= \mathbb{E}_\rho \left[ \frac{\sum_{j=1}^J L(\theta^j) f(\theta^j)}{\sum_{j=1}^J L(\theta^j)} - \mathbb{E}_{\rho^*}[f] \right] \sqrt{\mathbb{E}_\rho \left[ \left(1 - \frac{1}{J} \sum_{j=1}^J \frac{L(\theta^j)}{Z}\right)^2 \right]}$$
$$\leq \sqrt{4 \frac{\chi^2 \rho^* \|\rho\| + 1}{J}} \sqrt{\frac{1}{J} \mathbb{E}_\rho \left[ \frac{L(\theta)^2}{Z^2} \right]}$$
$$= 2 \frac{\chi^2 \rho^* \|\rho\| + 1}{J}$$

## 练习 (Rosenbrock 函数)

我们要采样的后验分布满足

$$\rho_{\text{post}}(\theta) \propto e^{-\Phi(\theta)} \rho_{\text{prior}}(\theta)$$
$$\Phi = \frac{1}{2} (y - G(\theta))^T \Sigma_y^{-1} (y - G(\theta))$$
$$= \frac{1}{2} (100(\theta_2 - c_1 \theta_1^2)^2 + (1 - \theta_1)^2)$$

```
In [43]: using Plots
using Random
using Distributions
using LinearAlgebra
Random.seed!(42)

function Phi_Rosenbrock(θ, c1, c2)
    return 100*(θ2 - c1*θ1^2)^2/c2 + (1.0 - θ1)^2/c2 + θ1^2/100 + θ2^2/100/2.0
end

function Phi_Rosenbrock(θ, c1, c2)
    return 100*(θ2 - c1*θ1^2)^2/c2 + (1.0 - θ1)^2/c2)/2.0
end

lx, lx = -6.0, 6.0
ly, ly = -4.0, 16
N = 5000
x = zeros(N, N)
y = zeros(N, N)
p = zeros(N, N)
for i = 1:N
    for j = 1:N
        x[i, j], y[i, j] = lx + (ux - lx) * ((1-1)/(N-1)), ly + (uy - ly) * ((j-1)/(N-1))
    end
end
dx = dy = 1/(N-1)

fig, ax = Plots.subplots(ncols=2, nrows=1, sharex=false, sharey=false, figsize=(8,3))
fig_error, ax_error = Plots.subplots(ncols=2, nrows=1, sharex=false, sharey=false, figsize=(8,3))

for k = 1:2
    cr, c1 = 10^(-2k-2.0), 1.0
    for j = 1:N
        for j = 1:N
            p[i, j] = Phi_Rosenbrock(x[i, j], y[i, j], c1, c2)
        end
    end

    p ./= exp(-p)
    z = sum(p)
    p ./= z
    p ./= (dx * dy)
    ax[k].contour(X, Y, p, 10)

    mean_ref = [sum(X.*p)/sum(p), sum(Y.*p)/sum(p)]
    @info "mean = ", mean_ref

    Js = [2^i for i=0:17]
    errors = zeros(length(Js))

    for (J, ind, J) in enumerate(Js)
        θ = 10*rand(Normal(0,1), J, 2)
        φ = zeros(J)
        for j = 1:J
            φ[j] = Phi_Rosenbrock(θ[j,1], θ[j,2], c1, c2)
        end
        φ .= minimum(φ)
        w = zeros(J)
        for j = 1:J
            w[j] = exp(-φ[j])
        end
        w ./= sum(w)
        mean_IS = (w*θ)[1]
        @info "mean estimated from importance sampling is ", mean_IS
        errors[J,ind] = norm(mean_IS - mean_ref)/norm(mean_ref)
    end

    ax_error[k].loglog(Js, errors, "o", label="Importance sampling")
    ax_error[k].loglog(Js, 1./sqrt(Js), "--", label="1/√J")
    ax_error[k].loglog(Js, 1./J, "--s", label="1/J")
    ax_error[k].set_ylabel("1/J")
    ax_error[k].set_xlabel("Rel. mean error")
    ax_error[k].legend()
end

fig.tight_layout()
fig.savefig("Rosenbrock_IS.pdf")

fig_error.tight_layout()
fig_error.savefig("Rosenbrock_IS_error.pdf")

[ Info: "mean = ", [0.9253911747282657, 1.7546848494609977]
[ Info: "mean estimated from importance sampling is ", [1.1536540161988076, 2.9427162674561673]
[ Info: "mean estimated from importance sampling is ", [1.6182524262125314, 3.3842079247340444]
[ Info: "mean estimated from importance sampling is ", [0.9740422857378545, 1.295450126480252]
[ Info: "mean estimated from importance sampling is ", [1.013929307615152, 1.952071707397683]
[ Info: "mean estimated from importance sampling is ", [0.626075016267751, 1.286943470945982]
[ Info: "mean estimated from importance sampling is ", [1.061162680081127, 1.8831355926008218]
[ Info: "mean estimated from importance sampling is ", [0.861356786272734, 1.727982871394913]
[ Info: "mean estimated from importance sampling is ", [0.92142284216867, 1.643727393438383]
[ Info: "mean estimated from importance sampling is ", [0.958023897313262, 1.7993644136934186]
[ Info: "mean = ", [0.990912653489645, 0.01970171189840336]
[ Info: "mean estimated from importance sampling is ", [1.475644788871807, -0.0994739950266997]
[ Info: "mean estimated from importance sampling is ", [0.448227519244443, 0.2212340463045559]
[ Info: "mean estimated from importance sampling is ", [0.2690468318133945, -0.06726089411892272]
[ Info: "mean estimated from importance sampling is ", [0.4215097604280212, -0.03171542947674044]
[ Info: "mean estimated from importance sampling is ", [1.10459236075503, 0.022540369792934]
[ Info: "mean estimated from importance sampling is ", [1.0122388378598166, 0.026074667656751337]
[ Info: "mean estimated from importance sampling is ", [0.8344618358416795, 0.0086031646462207]
[ Info: "mean estimated from importance sampling is ", [0.9151672058318165, 0.01061654085653156]
[ Info: "mean estimated from importance sampling is ", [1.0068544131180728, 0.01643190326619009]

15.0
12.5
10.0
7.5
5.0
2.5
-2.5
-8 -6 -4 -2 0 2 4 6 8

15.0
12.5
10.0
7.5
5.0
2.5
-2.5
-8 -6 -4 -2 0 2 4 6 8

10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^3 10^4 10^5

10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^3 10^4 10^5

3. 无迹变换
```

对于无迹变换, 对  $1 \leq j \leq N_s$

$$W_j^m = W_{j,N_s}^m \quad W_j^c = W_{j,N_s}^c = \frac{1}{2c_j^2} \sum_{i=1}^{2N_s} W_i^m = 1$$

使用Taylor展开, 我们有

$$G(\theta) = G(m) + \nabla G \delta\theta + \frac{1}{2} \nabla^2 G \delta\theta \otimes \delta\theta + \frac{1}{6} \nabla^3 G \delta\theta \otimes \delta\theta \otimes \delta\theta + O(\delta\theta^4)$$
$$\mathbb{E}[G(\theta)] = G(m) + \frac{1}{2} \nabla^2 G C + O(\delta\theta^4)$$
$$\text{Cov}[G(\theta), G(\theta)] = \mathbb{E} \left[ \left( \nabla G \delta\theta + \frac{1}{2} \nabla^2 G \delta\theta \otimes \delta\theta + \frac{1}{6} \nabla^3 G \delta\theta \otimes \delta\theta \otimes \delta\theta - \frac{1}{2} \nabla^2 G C + O(\delta\theta^4) \right) \right. \\ \left. \left( \nabla G \delta\theta + \frac{1}{2} \nabla^2 G \delta\theta \otimes \delta\theta + \frac{1}{6} \nabla^3 G \delta\theta \otimes \delta\theta \otimes \delta\theta - \frac{1}{2} \nabla^2 G C + O(\delta\theta^4) \right)^T \right]$$
$$= \nabla G C \nabla G^T + O(\delta\theta^4)$$

其中  $\nabla^2 G$  是在  $m$  的取值,  $\delta\theta = \theta - m$ , 我们认为  $C = O(\delta\theta^2)$ , 对于期望的近似, 我们有

$$\widehat{\mathbb{E}}[G] = W_0^m G(m) + \sum_{j=1}^{N_s} (W_j^m G(m) + c_j \nabla G[\sqrt{C}]_j + W_{j,N_s}^m G(m - c_j \sqrt{C}]_j)$$
$$= W_0^m G(m) + \sum_{j=1}^{N_s} (W_j^m G(m) + c_j \nabla G[\sqrt{C}]_j + \frac{c_j^2}{2} \nabla^2 G[\sqrt{C}]_j \otimes [\sqrt{C}]_j + \frac{c_j^3}{6} \nabla^3 G[\sqrt{C}]_j \otimes [\sqrt{C}]_j \otimes [\sqrt{C}]_j + O(\delta\theta^4))$$
$$+ W_{j,N_s}^m (G(m) - c_j \nabla G[\sqrt{C}]_j + \frac{c_j^2}{2} \nabla^2 G[\sqrt{C}]_j \otimes [\sqrt{C}]_j - \frac{c_j^3}{6} \nabla^3 G[\sqrt{C}]_j \otimes [\sqrt{C}]_j \otimes [\sqrt{C}]_j + O(\delta\theta^4))$$
$$= G(m) + \sum_{j=1}^{N_s} \frac{c_j^2}{2} (W_j^m + W_{j,N_s}^m) \nabla^2 G[\sqrt{C}]_j \otimes [\sqrt{C}]_j + O(\delta\theta^4) \quad \text{使用 } W_j^m = W_{j,N_s}^m$$

定义  $P = \sum_{j=1}^{N_s} \frac{c_j^2}{2} (W_j^m + W_{j,N_s}^m) [\sqrt{C}]_j \otimes [\sqrt{C}]_j$ , 对于方差的近似, 我们有

$$W_j^c (G_j(m) - \widehat{\mathbb{E}}[G_j]) (G_j(m) - \widehat{\mathbb{E}}[G_j])^T + \sum_{j=1}^{N_s} W_j^c (G_j(\theta) - \widehat{\mathbb{E}}[G_j]) (G_j(\theta) - \widehat{\mathbb{E}}[G_j])^T + W_{j,N_s}^c (G_j(\theta) - \widehat{\mathbb{E}}[G_j]) (G_j(\theta) - \widehat{\mathbb{E}}[G_j])^T$$
$$= W_0^c \nabla^2 G P \nabla^2 G^T + \sum_{j=1}^{N_s} \left( W_j^c (c_j \nabla G[\sqrt{C}]_j + \nabla^2 G[\frac{c_j^2}{2} \sqrt{C}]_j \otimes [\sqrt{C}]_j - P) (c_j \nabla G[\sqrt{C}]_j + \nabla^2 G[\frac{c_j^2}{2} \sqrt{C}]_j \otimes [\sqrt{C}]_j - P) \right)^T$$
$$+ W_{j,N_s}^c (c_j \nabla G[\sqrt{C}]_j + \nabla^2 G[\frac{c_j^2}{2} \sqrt{C}]_j \otimes [\sqrt{C}]_j - P) (-c_j \nabla G[\sqrt{C}]_j + \nabla^2 G[\frac{c_j^2}{2} \sqrt{C}]_j \otimes [\sqrt{C}]_j - P)^T + O(\delta\theta^4)$$
$$= \sum_{j=1}^{N_s} 2c_j^2 W_j^c \nabla G[\sqrt{C}]_j \otimes [\sqrt{C}]_j \nabla G^T + O(\delta\theta^4) \quad \text{使用 } W_j^m = W_{j,N_s}^m$$
$$= \nabla G C \nabla G^T + O(\delta\theta^4) \quad \text{使用 } W_j^c = W_{j,N_s}^c = \frac{1}{2c_j^2}$$

## 4. 卡尔曼变换

```
In [33]: using Random, Distributions, LinearAlgebra, Plots

function Gaussian_2d(μ_mean::Array{PT,1}, θ0_cov::Nx1{IT}, Ny::IT, x_min=nothing, x_max=nothing, y_min=nothing, y_max=nothing) where {PT<:AbstractFloat, IT<:Int}
    # 2d Gaussian plot

    if x_min == nothing
        x_min = μ_mean[1] + 5*sqrt(θ0_cov[1,1])
    end
    if x_max == nothing
        x_max = μ_mean[1] + 5*sqrt(θ0_cov[1,1])
    end
    if y_min == nothing
        y_min = μ_mean[2] + 5*sqrt(θ0_cov[2,2])
    end
    if y_max == nothing
        y_max = μ_mean[2] + 5*sqrt(θ0_cov[2,2])
    end

    xx = Array{LinRange(x_min, x_max, Nx)}
    yy = Array{LinRange(y_min, y_max, Ny)}
    X, Y = repeat(xx, 1, Ny), repeat(yy, 1, Nx)
    Z = zeros{PT, Nx, Ny}

    det_θ0_cov = det(θ0_cov)

    for ix = 1:Nx
        for iy = 1:Ny
            Δxy = [xx[ix] - μ_mean[1], yy[iy] - μ_mean[2]]
            Z[ix, iy] = exp(-0.5*(Δxy'θ0_covΔxy)) / (2 * pi * sqrt(det_θ0_cov))
        end
    end

    return X, Y, Z
end

function EKRF(G::Function, m::Array{Float64,1}, C::Array{Float64,2})
    g = G(m)
    mg, Cg = g, C+dg
    return mg, Cg
end

function UKF(G::Function, m::Array{Float64,1}, C::Array{Float64,2}, J::Int64)
    N_θ = length(m)
    N_ens = 2N_θ + 1

    # weights
    a = min(sqrt(4/(N_θ + 1)), 1.0)
    λ = a*(N_θ + 1) - N_θ
    sigma_weight = sqrt(N_θ + λ)
    cov_weight = 1/(2(N_θ + λ))

    # generate sigma points
    chol_C = cholesky(Hermitian(C))
    θ = zeros{Float64, N_θ, N_ens}
    θ[:, 1] = m
    for i = 1:N_θ
        θ[:, i+1+N_θ] = m + sigma_weight*chol_C[:,i]
        θ[:, i+1-N_θ] = m - sigma_weight*chol_C[:,i]
    end

    g = zeros{Float64, N_θ, N_ens}
    for i = 1:2N_θ+1
        g[i, i+N_θ] = G(θ[:, i])
    end

    # compute mean
    mg = g[:, 1]

    Cg = zeros{Float64, N_θ, N_θ}
    for i = 1:N_ens
        Cg += cov_weight*(g[i, i+N_θ] - mg)*(g[i, i+N_θ] - mg)'
    end

    return mg, Cg
end

function EKRF(G::Function, m::Array{Float64,1}, C::Array{Float64,2}, J::Int64)
    θ = rand(NvNormal(m, C), J)
    g = zeros{Float64, N_θ, J}
    for i = 1:J
        g[i, 1] = G(θ[:, i])
    end

    mg = sum(g, dims=2)/J
    Cg = zeros{Float64, N_θ, N_θ}
    for i = 1:J
        Cg += (g[i, 1] - mg)*(g[i, 1] - mg)'
    end
    Cg ./= (J - 1)

    return mg, Cg, g
end

function G(θ)
    g = [1 + sqrt(θ[1]^2 + θ[2]^2); exp(θ[1]/2) * θ[2]^2]
    dg = [θ[1]/(sqrt(1+θ[1]^2 + θ[2]^2)) θ[2]/(2*sqrt(1+θ[1]^2 + θ[2]^2)); θ[2]/(2*sqrt(1+θ[1]^2 + θ[2]^2)) θ[1]^2/2]
    return g, dg
end

Out[3]: EKRF (generic function with 1 method)
```

```
In [34]: m, C = [10.0; 10.0], [1.0^2 0.0; 0.0 1.0^2]

mckf = EKRF(G, m, C, 100000)
ekkf = EKRF(G, m, C)
ukf = UKF(G, m, C)
ekf = EKRF(G, m, C, 50)

Nx = Ny = 100

fig, ax = Plots.subplots(ncols=4, sharex=false, sharey=true, figsize=(12,3))
for i = 1:4
    ax[i].scatter(mckf[3][1, i], mckf[3][2, i], color="grey", alpha=0.5, label="MCMC")
end

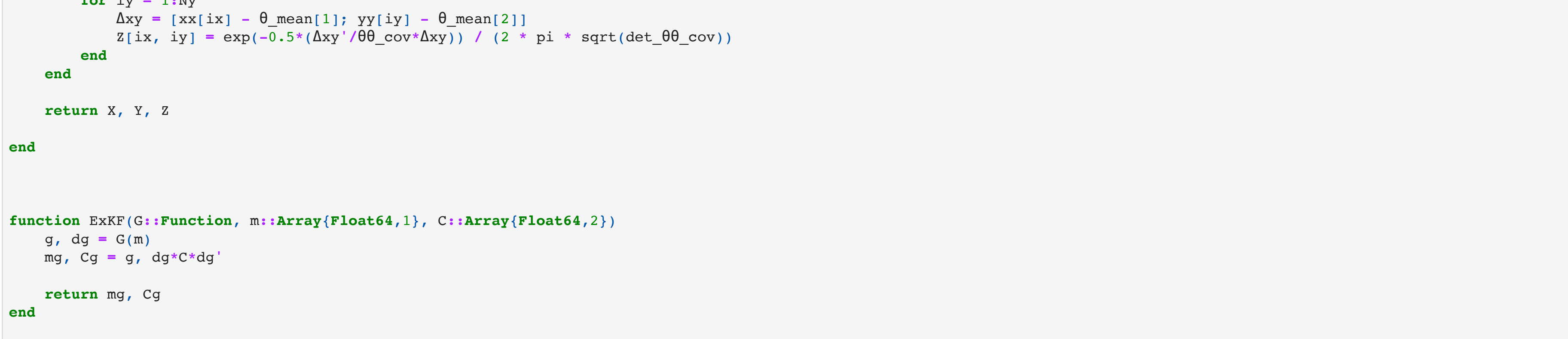
X, Y, Z = Gaussian_2d(ekkf[1], ekkf[2], Nx, Ny)
ax[1].contour(X, Y, Z, 50, colors="red", alpha=0.5, label="EKRF (J=1)")
ax[1].set_title("EKRF (J=1)")

X, Y, Z = Gaussian_2d(ukf[1], ukf[2], Nx, Ny)
ax[2].contour(X, Y, Z, 50, colors="red", alpha=0.5, label="UKF (J=5)")
ax[2].set_title("UKF (J=5)")

ax[3].scatter(ekkf[3][1, i], ekkf[3][2, i], color="red", label="EKRF")
ax[3].set_title("EKRF (J=50)")

ax[4].set_title("MCMC (J=10^5)")

fig.tight_layout()
fig.savefig("XK-1.png")
```



```
In [35]: m, C = [1.0; 1.0], [1.0^2 0.0; 0.0 1.0^2]

mckf = EKRF(G, m, C, 100000)
ekkf = EKRF(G, m, C)
ukf = UKF(G, m, C)
ekf = EKRF(G, m, C, 50)

Nx = Ny = 100

fig, ax = Plots.subplots(ncols=4, sharex=false, sharey=true, figsize=(12,3))
for i = 1:4
    ax[i].scatter(mckf[3][1, i], mckf[3][2, i], color="grey", alpha=0.5, label="MCMC")
end

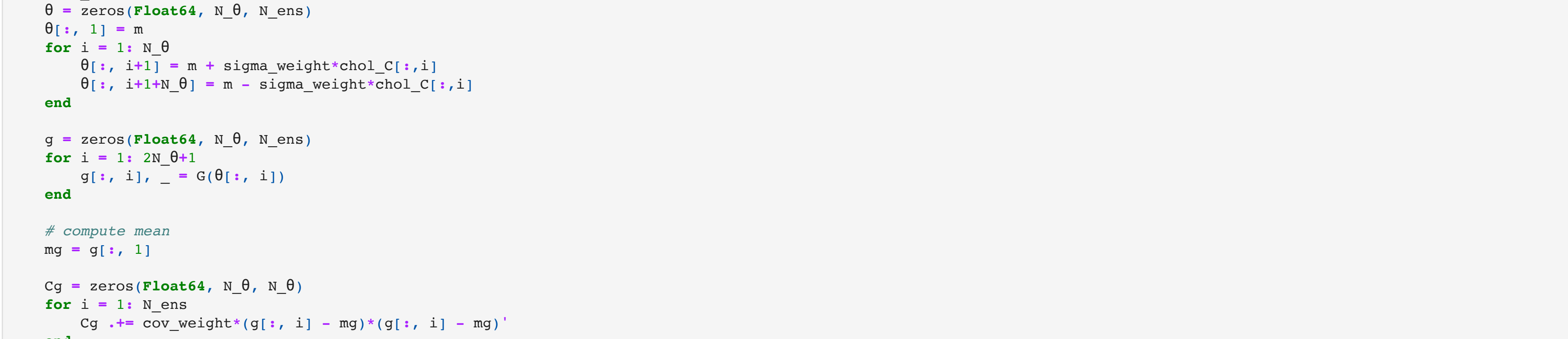
X, Y, Z = Gaussian_2d(ekkf[1], ekkf[2], Nx, Ny)
ax[1].contour(X, Y, Z, 50, colors="red", alpha=0.5, label="EKRF (J=1)")
ax[1].set_title("EKRF (J=1)")

X, Y, Z = Gaussian_2d(ukf[1], ukf[2], Nx, Ny)
ax[2].contour(X, Y, Z, 50, colors="red", alpha=0.5, label="UKF (J=5)")
ax[2].set_title("UKF (J=5)")

ax[3].scatter(ekkf[3][1, i], ekkf[3][2, i], color="red", label="EKRF")
ax[3].set_title("EKRF (J=50)")

ax[4].set_title("MCMC (J=10^5)")

fig.tight_layout()
fig.savefig("XK-2.png")
```



## 5. 实值非体积保持模型

对于仿射耦合

$$y_{1:d} = x_{1:d}$$
$$y_{d+1:N_s} = \mathcal{Z}_{d+1:N_s} \odot \exp(\epsilon(x_{1:d})) + t(x_{1:d})$$

可逆

$$\mathcal{Z}_{1:d} = y_{1:d}$$
$$\mathcal{Z}_{d+1:N_s} = \exp(-\epsilon(x_{1:d})) \odot (y_{d+1:N_s} - t(x_{1:d}))$$

微分熵

$$\frac{\partial \theta y}{\partial x} = \begin{bmatrix} I & 0 \\ \frac{\partial \epsilon}{\partial x_{1:d}} & \exp(\epsilon(x_{1:d})) \end{bmatrix}$$