HOW TO REALIZE THE GEODESIC OVER $\mathbb{L}^2$ WITH SRVF

H.LUO

(1) Main idea
Choose a specific representation of p.d.f.s such that the Fisher-Rao metric has the feature of
  (a) Coordinate independent
  (b) Re-parameterization invariant⇔Invariant under diffeomorphism group
  (c) Calculable
  Setup: $\mathcal{C}_1\, and\, \mathcal{C}_2$ are probably the same curve under different parameterizations. How can we tell whether they are the same?
  $Geodesics\, between\, \mathcal{C}_1\, and\, \mathcal{C}_2 \to Moments \to Normal\, approximation$[1] $\overset{exponential}{\to} Approximation\, to\, \mathcal{C}_1\, and\, \mathcal{C}_2 \to Have\, similar\, approximations$(hopefully)
    (i) Geodesic distance: Use SRVF representation to yield an explicit form for exponential mapping.
    (ii) Moments: Use Karcher mean and covariance to classify 3-dim curves.
    (iii) Simulation: Use Gaussian with corresponding Karcher mean and covariance to sample from true model that generates the curves(points). Nothing new, probably we can even compile a Bayesian model here...

(2) Representations
For a 3-dimensional curve, $\beta : [0,1] \to \mathbb{R}^3$
  (a) SRF $h(t) := \sqrt{|\beta'(t)|}\beta(t)$
  (b) SRVF $q(t) := F(v(t)) = \begin{cases} \frac{v}{\sqrt{|v|}} & |v| \neq 0 \\ 0 & otherwise \end{cases}$ where $v(t) = \beta'(t)$ is the velocity vector along the curve β[2]
  The action of $SO(3)$ is by composition; the action of $\mathcal{D}^{\infty}_{\mathbb{S}^0}$ is by $\gamma(\beta) = \beta \circ \gamma\sqrt{\gamma'}$. These two actions induce isometries and commutes. We can consider the spanning group of these two collection of actions and called it Rigid-Reparameterization group[3].
  (c) The SRF and SRVF spaces are subsets of $\mathbb{L}^2$ which is square-integrable curves[4].
  Which to use depends on what kind of features we want to analyze.
    (i) For the shape analysis of curves: SRVF

Date: 06-09-2016.

[1] How can we calculate the moments from geodesic? Notice that moments are only tangent velocity vectors, so what we have to do is to calculate the velocity vector field along the geodesic and its speed and accelerate are corresponding 1st and 2nd moments.

[2] Like a normalized velocity vector field.

[3] However, the $\mathcal{D}^{\infty}_{\mathbb{S}^0}$ is complete yet $\mathcal{D}^{\infty}_{\mathbb{L}^2}$ is not, because $\mathbb{L}^2$ is not compact.

[4] Counterexample, space-filling curve.

  - (A) Translation invariant, i.e. invariant under $\tau_{\mathbb{R}^3}$.
  - (B) Inverse function problem. The solution is algebraically simple$\beta(t) := \int_0^t q(s)|q(s)|ds, t \in [0,1]$.
- (ii) For the position analysis and time-wrapping we discussed: SRF along with the *path-straightening method* to calculate the geodesics between two curves.
  - (A) The square-root function space is not closed in $\mathbb{L}^2$ either.
  - (B) Use the induced frame of $\mathbb{L}^2$ to make this into a Riemannian manifold.
  - (C) The tangent vector for a certain point β is $dh_\beta(p) = \frac{d}{dr}h_\beta(\beta + rp)|_{r=0} = \frac{1}{2\sqrt{|\beta'|}}\beta' p' \beta + \sqrt{|\beta'|}p$.
  - (D) Use the alternative definition of geodesic that it is the energy-minimizing curve connecting two points.
    So the *path-straightening method* is essentially the rapidest descendence method in solving critical point of the energy function among the class of curves with endpoints fixed as two curves $\mathcal{C}_1\, and\, \mathcal{C}_2$.

(d) Fisher-Rao metric under SRVF: elastic metric[5]

$\langle q_1, q_2\rangle_q = a\int_0^1 \frac{1}{\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|}\delta\left|\frac{\partial\left[\int_0^t q_1(s)|q_1(s)|ds\right]}{\partial t}\right|\delta\left|\frac{\partial\left[\int_0^t q_2(s)|q_2(s)|ds\right]}{\partial t}\right| +$

$b\int_0^1\left[\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|\right]\left\langle \delta\frac{\frac{\partial\left[\int_0^t q_1(s)|q_1(s)|ds\right]}{\partial t}}{\left|\frac{\partial\left[\int_0^t q_1(s)|q_1(s)|ds\right]}{\partial t}\right|}, \delta\frac{\frac{\partial\left[\int_0^t q_2(s)|q_2(s)|ds\right]}{\partial t}}{\left|\frac{\partial\left[\int_0^t q_2(s)|q_2(s)|ds\right]}{\partial t}\right|}\right\rangle_{\mathbb{L}^2}$

where $\delta q := \frac{\delta\int_0^t q(s)|q(s)|ds}{2\sqrt{\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|}}\frac{\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}}{\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|} + \sqrt{\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|}\delta\frac{\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}}{\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|}$

$q, q_1, q_2 \in SRVF$

(e) Unit Length curve in $\mathbb{L}^2$ is $\mathbb{S}^\infty$

The geodesic connecting f_1, f_2 on this hypersphere can be calculated via $\tau f_1 + (1-\tau)f_2$ and rescale to unit length.

$\psi(\tau) := \frac{1}{sin\theta}\left[sin(\theta(1-\tau))f_1 + sin(\theta\tau)f_2\right], \theta = arccos\langle f_1, f_2\rangle_{\mathbb{L}^2}$

$exp_f(v) = cos(\|v\|)f + sin(\|v\|)\frac{v}{\|v\|}$

$g \Longleftrightarrow v$

$exp_f^{-1}(g) = \frac{\theta}{sin\theta}\left[g - fcos\theta\right]$

[5] $p(t) := \left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|$ measures speed; $\theta(t) := \frac{\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}}{\left|\frac{\partial\left[\int_0^t q(s)|q(s)|ds\right]}{\partial t}\right|}$ measures direction.

(3) Feature spaces

Table 1. Description of different feature spaces for the SRVF representation

Features	Shape + orientation + scale	Shape + scale
Prespace	$\mathbb{L}^2$	$\mathbb{L}^2$
Equivalence classes	$[q] = \text{closure}\{(q,\gamma) \mid \gamma \in \Gamma\}$	$[q] = \text{closure}\{O(q,\gamma) \mid O \in SO(3),\ \gamma \in \Gamma\}$
Feature space	$\mathcal{S}_1 = \{[q] \mid q \in \mathbb{L}^2\}$	$\mathcal{S}_2 = \{[q] \mid q \in \mathbb{L}^2\}$
Distances	$d_1([q_1],[q_2]) = \min_{\gamma \in \Gamma} \lVert q_1 - (q_2,\gamma) \rVert$	$d_2([q_1],[q_2]) = \min_{O \in SO(3),\ \gamma \in \Gamma} \lVert q_1 - O(q_2,\gamma) \rVert$
Geodesics	Equation (3) between q_1 and q_2^*	Equation (3) between q_1 and q_2^*
Features	**Shape + orientation**	**Shape**
Prespace	$\mathbb{S}_\infty$	$\mathbb{S}_\infty$
Equivalence classes	$[q] = \text{closure}\{(q,\gamma) \mid \gamma \in \Gamma\}$	$[q] = \text{closure}\{O(q,\gamma) \mid O \in SO(3),\ \gamma \in \Gamma\}$
Feature space	$\mathcal{S}_3 = \{[q] \mid q \in \mathbb{S}_\infty\}$	$\mathcal{S}_4 = \{[q] \mid q \in \mathbb{S}_\infty\}$
Distances	$d_3([q_1],[q_2]) = \min_{\gamma \in \Gamma} \cos^{-1}(\langle q_1,(q_2,\gamma)\rangle)$	$d_4([q_1],[q_2]) = \min_{O \in SO(3),\ \gamma \in \Gamma} \cos^{-1}(\langle q_1, O(q_2,\gamma)\rangle)$
Geodesics	Equation (4) between q_1 and q_2^*	Equation (4) between q_1 and q_2^*

[6]

If we allow no scaling, then we are one `Shape` + `orientation` with the represented space $\mathbb{S}^\infty$. Attention that to override the $\mathcal{D}^\infty_{\mathbb{L}^2}$ not being complete, we proposed the closure.

(4) Mean computation: "representative of observations"
Problem: capture variability associated with curves within object classes using probability models. i.e. Learn the model from the sample.
Solution: Use the Karcher mean $argmin_{[q]\in\mathcal{S}} \sum_i geodesic\,(q, q_i)$ to classify which equivalent class this set of observed curves represented in SRVF form$\{q_1, \cdots q_n\}$ should belong to. This is like a k-means method except that the distance is chosen to be geodesic distance over a sphere. The major computation time may be due to step 1 and 3 in Alg.1 on p.1158.
If we need a "typical" curve of this equivalent class, we can use extrinsic/intrinsic[7] average of points$(exp^{-1}(curves))$ and project it back to the feature space.

(5) Probability models on space of shapes
Consider the distortion of shapes of curves as noise generated by a truncated Gaussian. It has to be truncated because it must be support in $[0,1]$ to be put into tangent space.

$$\begin{cases} computational & \begin{cases} parametric & [Kurtek\,et.al] \\ nonparametric & Kernel\,estimation \end{cases} \\ embedding & \begin{cases} intrinsic \\ extrinsic \end{cases} \end{cases}$$

Then we can use some classical technique like comparison of log-likelihoods to classify the proteins.

(6) Conclusion
It sounds a good idea to proceed in this method, by using truncated Gaussians as "generic form" for a collection of curves known to have some common properties. Another contribution is that this paper provided a I think the most time-consuming part may be the calculation of Karcher mean when there are a lot of samples.

[6][Kurtek et.al]
[7]Leave one out average.

(7) Questions

The "...the space of diffeomorphisms is not closed" on p.1156 should be "...not complete".

Another puzzling question is that in 5.2 of [Kurtek et.al], the author said that "traditional PCA" can be used for selecting an efficient basis. Since this PCA is done directly on the observed tangential matrix, will it be highly biased in a certain direction[8]? I mean if $\mathcal{S}$ is some quotient space without scaling feature, should this PCA be modified correspondingly? The same concern applied to p.1161.

An interesting question to explore is that how will the sampling random curves be if we yield the model from $\mathcal{S}_4$ yet use the exponential mapping in $\mathcal{S}_3$ to map them back?

One place needs to be explained further, theoretically, is that why "...if the greatest eigenvalue of the covariance matrix becomes large, a unimodal density in the tangent space may map to a bimodal density.". This claim is not clear to me, especially when the $k \to \infty$.

[8]For example, if a certain direction is wildly rapidly oscillating then PCA may give weird results. Since PCA is scale-dependent, should we consider normalized tangent matrix instead? Or alternatively, Gram-Schmidt method is the first choice when I consider the problem of choosing a basis here.

References

[Kurtek et.al] Kurtek, Sebastian, et al. "Statistical modeling of curves using shapes and related features." Journal of the American Statistical Association 107.499 (2012): 1152-1165.

```
#Reference
#[S.Kurtek et.al]Statistical Modeling of Curves Using Shapes and
Related Features, 2012.
#[J.Baek et.al]Finding Geodesics on Surfaces, 2007
library(cubature)
```

```
## Warning: package 'cubature' was built under R version 3.1.3
```

```
library(numDeriv)
```

```
## Warning: package 'numDeriv' was built under R version 3.1.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
#Riemannian metric function g
g<-function(v1,v2){
  #Treat two points on $S^2$ as endpoints of 2 vectors. The
gradient at these endpoints are exactly in the tangent space of
these points.
  integrand1 <- function(x) {v1(x)*v2(x)}
  innerprod <- adaptIntegrate(integrand1, 0, 1, tol=1e-4)
  return(innerprod$integral)
}
#Test metric
#g(p_1,p_2)


#If two pdfs are regarded as points on the $S^{n}$ sphere, then
all we have to do is to parameterize the sphere and figure out the
lesser arc connecting them because under the SRF representation,
the space of $\mathcal{P}$ becomes $\mathbb{L}^2$ sphere and the
metric becomes a global one. That is to say we can directly
realize the geodesic analytically and with the same
parameterization we can find out the pdfs along this geodesic.

#To choose such a parameterization and not dealing with the
infinite dimensional space directly, let us consider the beta
family parameterized with the classical parameters.


#If the Riemannian metric changes along the geodesic, then one
thing is we have to include extra terms into the g;the second
thing is we have to recalculate the general expression of geodesic
under such an inner product. But unlike $S^2$, there might not be
a global expression for geodesic on such a manifold.


#However, if we take the definition of the geodesic and try to
carry out a variational calculation to get the geodesic, the
difficulty arise in that we do not yield an ODE but a functional
equation which requires additional work to solve by the classical
fixed point theorem algorithm.
#arc length between two points, since they are on the unital
sphere.
#acos(g(dp_1,dp_2))
#angle between two points on an arc should be: k*\phi, k\in[0,1],
and now we take two point vectors as basis.
get_arc<-function(endpoint1,endpoint2,x,k){
  #We are doing the arc from endpoint1 to endpoint2.
  #rescale them into srf form.
  srf1<-function(x){sqrt(endpoint1(x))}
  srf2<-function(x){sqrt(endpoint2(x))}
  v1<-function(x){grad(srf1,x)}
  v2<-function(x){grad(srf2,x)}
  #k is the scaling parameter which lying in [0,1]
  #phi<-acos(g(v1,v2)/sqrt(g(v1,v1)*g(v2,v2)))
  phi<-acos(g(srf1,srf2))
  message("angle=",phi)
  #So we calculate the vector first and then rescale it to one.
  #result1<-function(x){srf1(x)*(1-k)+srf2(x)*(k)}
  #result2<-result1(x)/g(result1,result1)
  #We can also use the analytic formula
```

```
  result1<-function(x){1/sin(phi) * ( sin(phi-k*phi)*srf1(x) +
sin(k*phi)*srf2(x) )}
  result2<-result1(x)
  return(result2^2)
}

#With a fixed k the returning value is a point on the arc
connecting point1 and point2 evaluated at the value x. k is a
parameter for the arc connecting point1 and point2.
#We write a plotting function.
get_arc_length<-function(point1=p_1,point2=p_2){
  integrand2<-function(x){
    get_arc(p_1,p_2,x[1],x[2])
  }
  adaptIntegrate(integrand2,c(0,0),c(1,1))
}
plot_geodesic<-
function(p1=p_1,p2=p_2,precision=10,osc=0,xl=5,yl=1){
  dev.new()

  colfunc <- colorRampPalette(c("red", "blue"))
  leng=precision
  leng.step=1
  colgrad=colfunc(leng)
  for( k in seq(from=osc,to=1-osc,length.out=leng) ){
    #we set osc to avoid infinity at 0 and 1 for some weired pdfs.
    fun_k<-function(x){get_arc(p1,p2,x,k)}
    plot(fun_k,xlim=c(-
xl,xl),ylim=c(0,yl),col=colgrad[leng.step],cex=2)
    leng.step=leng.step+1
    par(new=TRUE)
    Sys.sleep(3/leng)
  }
}
```

```
#Some examples
dtgaussian<-function(x,mu,sigma){
#truncated gaussian on [0,1]
  return(dnorm(x,mu,sigma)/(pnorm(1,mu,sigma)-pnorm(0,mu,sigma))*
(x>=0)*(x<=1))
}
p_1<-function(x){
  return(dtgaussian(x,0,1))
}
p_2<-function(x){
  return(dtgaussian(x,5,1))
}
p_3<-function(x){
  return(dbeta(x,1,10))
}
p_4<-function(x){
  return(dbeta(x,2,6)*0.25+dbeta(x,6,2)*0.25)
}
p_5<-function(x){
  return(dbeta(x,4,9))
}

plot_geodesic(p_1,p_2,precision=10,osc=0,yl=5)
```

```
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
## angle=0.620740233728275
```

```
plot_geodesic(p_3,p_4,precision=10,osc=0,yl=5)
```

```
## angle=1.16838972406597
```

```
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
## angle=1.16838972406597
```

```
plot_geodesic(p_3,p_5,precision=10,osc=0,yl=10)
```

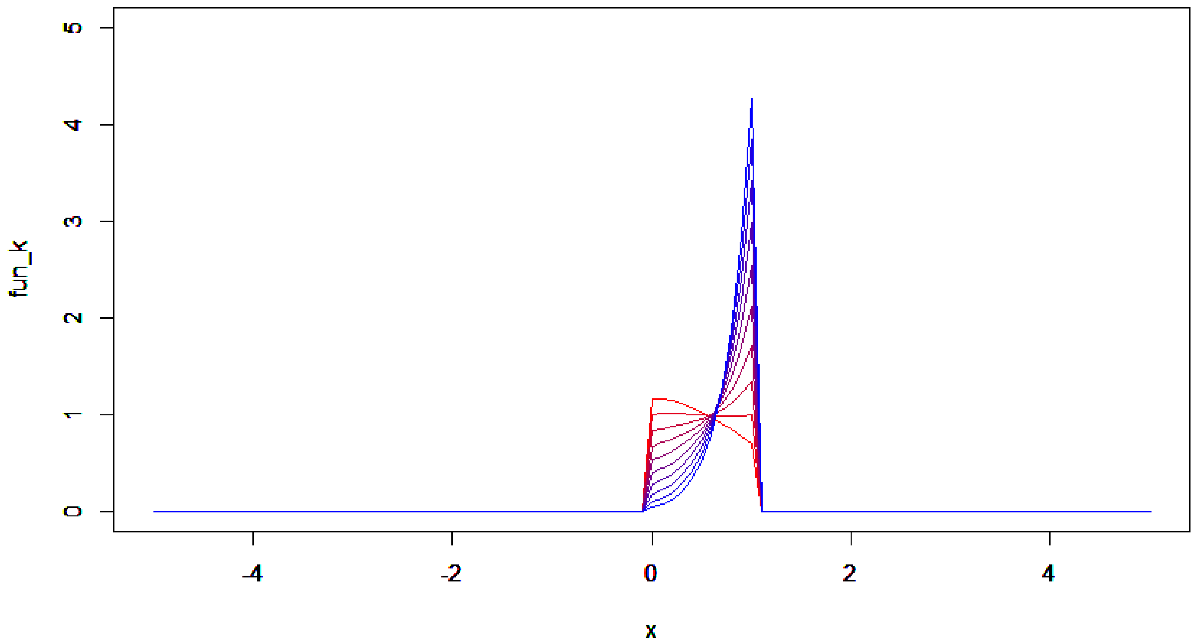
```
## angle=0.977594908991111
```

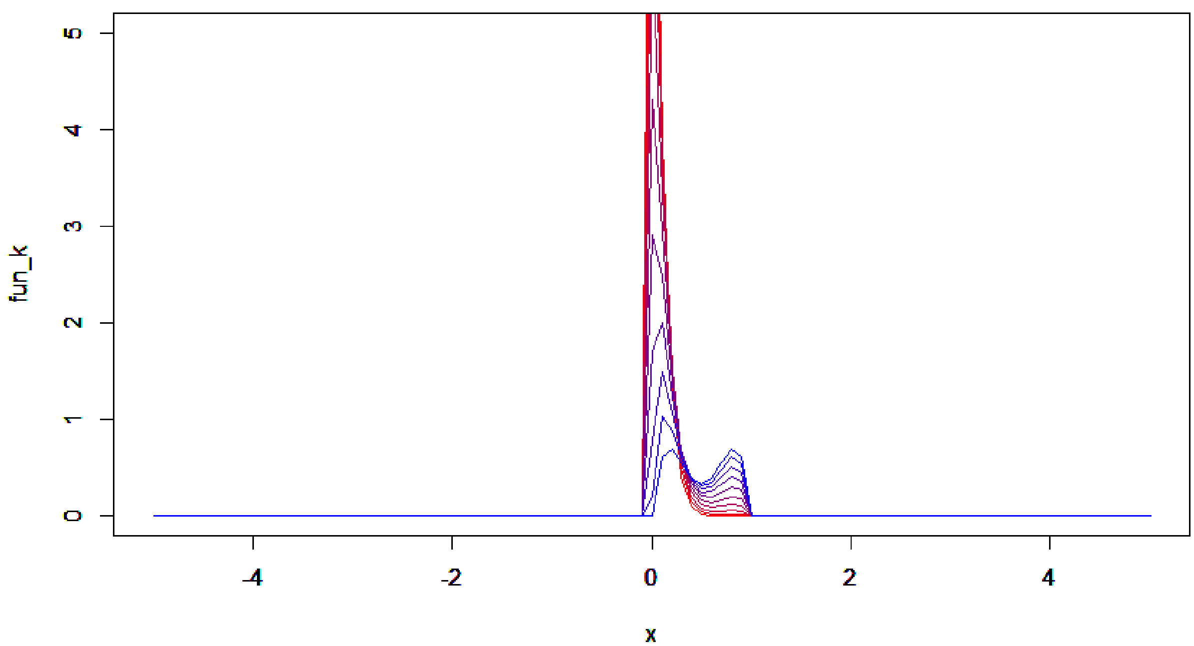
```
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
## angle=0.977594908991111
```

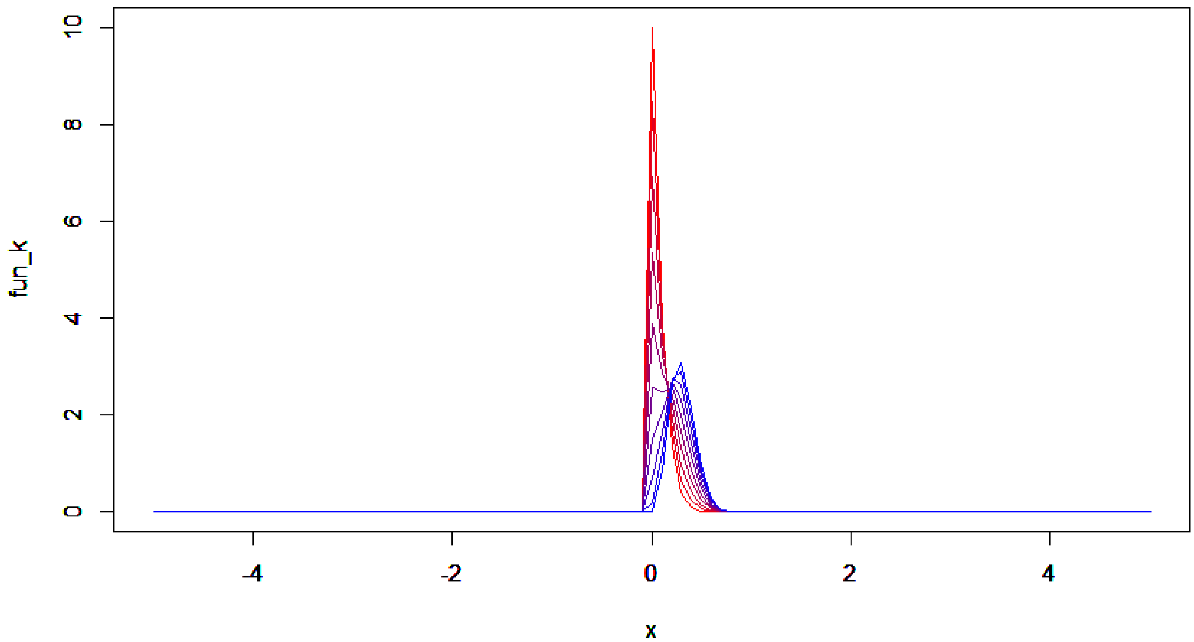
```
plot_geodesic(p_1,p_4,precision=10,osc=0,yl=2,xl=1)
```

```
## angle=0.804252092604058
```

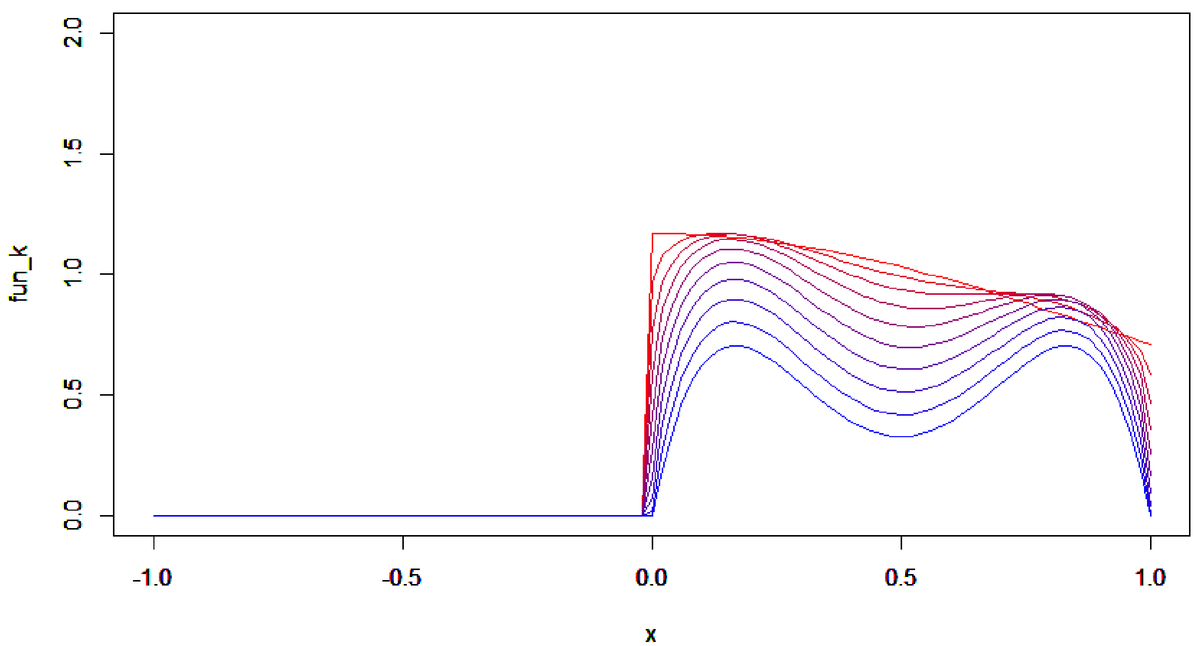
```
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
## angle=0.804252092604058
```


fun_k
x
-4
-2
0
2
4
0
1
2
3
4
5


fun_k
x
-4
-2
0
2
4
0
1
2
3
4
5



fun_k
0
2
4
6
8
10
-4
-2
0
2
4
x



fun_k
0.0
0.5
1.0
1.5
2.0
-1.0
-0.5
0.0
0.5
1.0
x