The Abdus Salam
International Centre
for Theoretical Physics

2470-5

Advanced School and Workshop on Matrix Geometries and Applications

1 - 12 July 2013

Computational issues of matrix geometric means (2)

Dario Bini
Università di Pisa
Italy

Computational issues of matrix geometric means
Part 2

D.A. Bini
Università di Pisa

Trieste – July 2013

Contents

- mean of $k > 2$ matrices: The ALM mean
- analysis of the convergence speed
- a mean defined in terms of medians
- a parametric class of means: analysis of the convergence speed
- implementation in Octave/Matlab
- numerical experiments and plots

The ALM mean

Remark

The ALM-properties uniquely define the geometric mean of two matrices A and B

For $k > 2$ matrices there exist infinitely many matrix means satisfying the ALM-properties

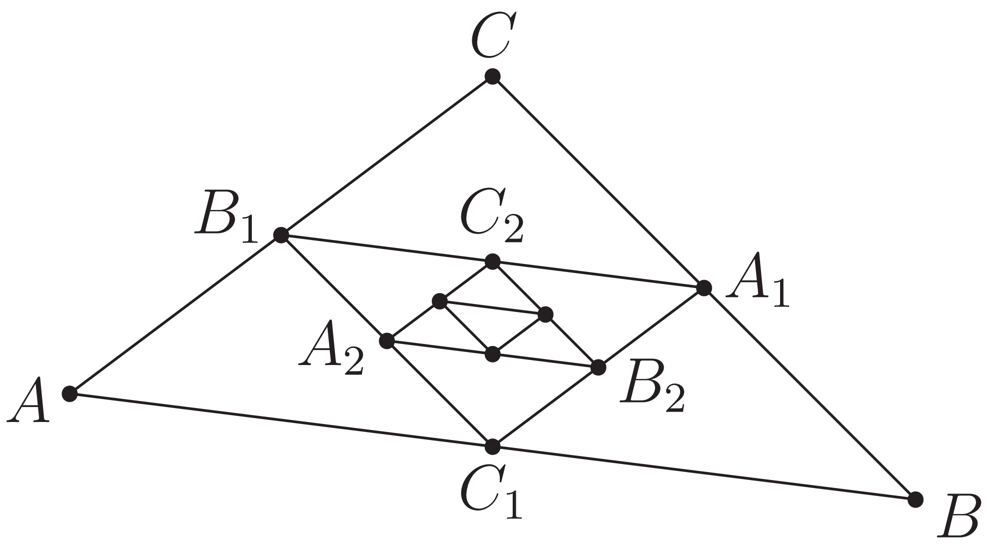
One of these means is the Ando–Li–Mathias (ALM) mean
The ALM mean [Ando–Li–Mathias, 2003]:

$$\begin{array}{lll} A_1 = G(B,C) & A_2 = G(B_1,C_1) & A_3 = G(B_2,C_2) \\ B_1 = G(C,A) & B_2 = G(C_1,A_1) & B_3 = G(C_2,A_2) \quad \ldots \\ C_1 = G(A,B) & C_2 = G(A_1,B_1) & C_3 = G(A_2,B_2) \end{array}$$

The three sequences have a common limit defined as the ALM mean $G_{ALM}(A,B,C)$

$$
\begin{array}{lll}
A_1 = G(B, C) & A_2 = G(B_1, C_1) & A_3 = G(B_2, C_2) \\
B_1 = G(C, A) & B_2 = G(C_1, A_1) & B_3 = G(C_2, A_2) \quad \ldots \\
C_1 = G(A, B) & C_2 = G(A_1, B_1) & C_3 = G(A_2, B_2)
\end{array}
$$

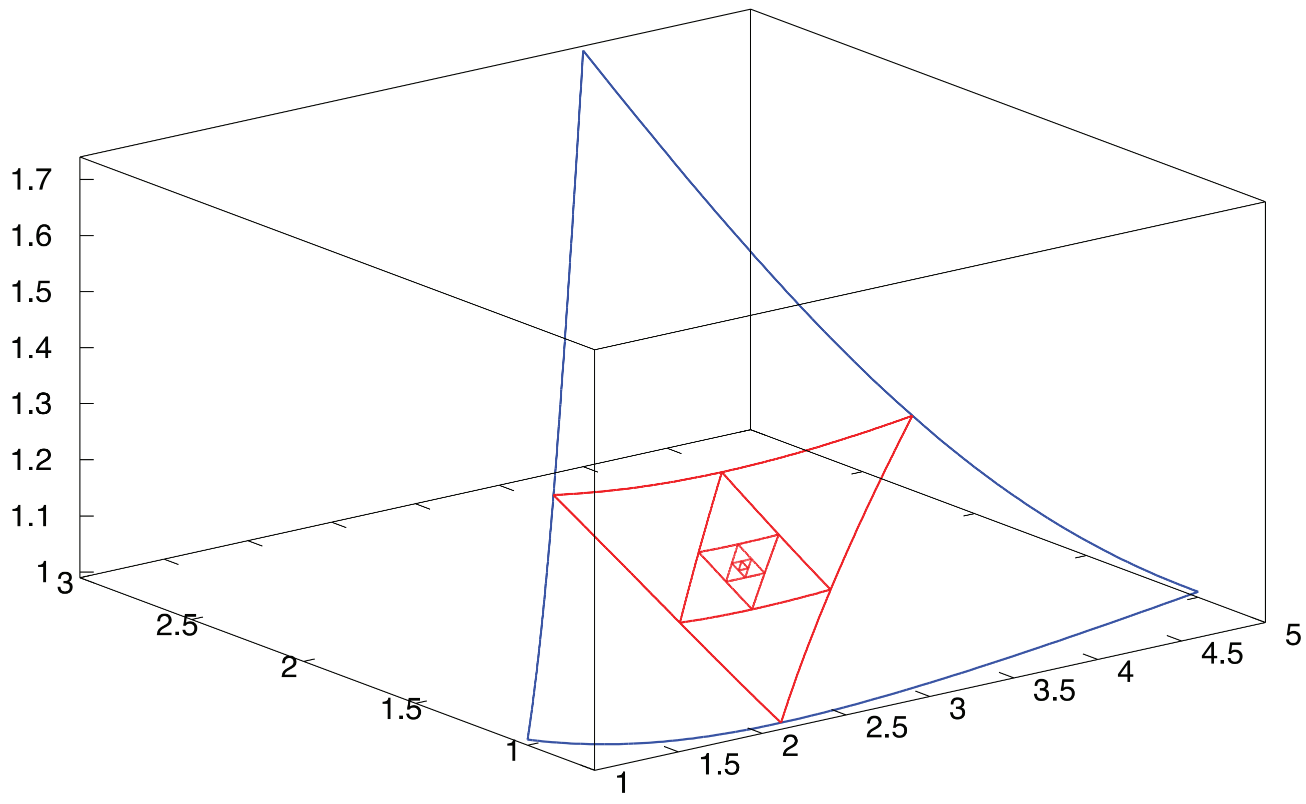
Recall that $C_1 = G(A, B)$ is the midpoint of the geodesic joining A and B. In the Euclidean geometry, this provides the following geometric interpretation



In the Euclidean geometry the sequences of the vertices converge to the centroid of the triangle ABC.

Computing the ALM mean

In the Riemannian geometry we have the following picture



Analysis of the convergence speed

Definition: The matrix sequence E_k linearly converges to L if

$$\lim_k \frac{\|E_{k+1} - L\|}{\|E_k - L\|} = \gamma$$

for some matrix norm $\|\cdot\|$ and for some $0 < \gamma < 1$

In this case there exists a constant α such that

$$\|E_k - L\| \le \alpha \gamma^k$$

Definition: The matrix sequence E_k converges to L superlinearly with order p if

$$\lim_k \frac{\|E_{k+1} - L\|}{\|E_k - L\|^p} = \gamma > 0$$

for some matrix norm $\|\cdot\|$ and for some $\gamma > 0$

In this case, there exist constants $\alpha, \beta > 0$, $0 < \lambda < 1$ such that

$$\|E_{k+1} - L\| \le \alpha \|E_k - L\|^p \qquad \|E_k - L\| \le \beta \lambda^{2^p}$$

Analysis of the convergence speed

Let $A, B, S \in \mathcal{P}_n$. Recall that $A\#_{\frac{1}{2}}B = A(A^{-1}B)^{\frac{1}{2}}$

Observe that $S(A\#_{\frac{1}{2}}B)S = (SAS)\#_{\frac{1}{2}}(SBS)$

If

$$SAS = I + E_A, \quad SBS = I + E_B, \quad \|E_A\|, \|E_B\| < \epsilon$$

Then

Lemma

$$\begin{aligned} S(A\#_{\frac{1}{2}}B)S =& (I + E_A)\left[(I + E_A)^{-1}(I + E_B)\right]^{\frac{1}{2}} \\ =& I + \frac{1}{2}E_A + \frac{1}{2}E_B + O(\epsilon^2) \end{aligned}$$

Proof. It follows from $(I + X)^{\frac{1}{2}} = I + \frac{1}{2}X + O(X^2)$

Therefore, if $S = G^{-\frac{1}{2}}$, $G := G(A, B, C)$,

$$\begin{aligned} A' &= B\#_{\frac{1}{2}}C & \qquad G^{-\frac{1}{2}}AG^{-\frac{1}{2}} &= I + E_A, \\ B' &= C\#_{\frac{1}{2}}A & \qquad G^{-\frac{1}{2}}BG^{-\frac{1}{2}} &= I + E_B \\ C' &= A\#_{\frac{1}{2}}B & \qquad G^{-\frac{1}{2}}CG^{-\frac{1}{2}} &= I + E_C \end{aligned}$$

then

$$\begin{aligned} E_{C'} &= G^{-\frac{1}{2}}C'G^{-\frac{1}{2}} - I = \tfrac{1}{2}E_A + \tfrac{1}{2}E_B + O(\epsilon^2) \\ E_{B'} &= G^{-\frac{1}{2}}B'G^{-\frac{1}{2}} - I = \tfrac{1}{2}E_A + \tfrac{1}{2}E_C + O(\epsilon^2) \\ E_{A'} &= G^{-\frac{1}{2}}A'G^{-\frac{1}{2}} - I = \tfrac{1}{2}E_B + \tfrac{1}{2}E_C + O(\epsilon^2) \end{aligned}$$

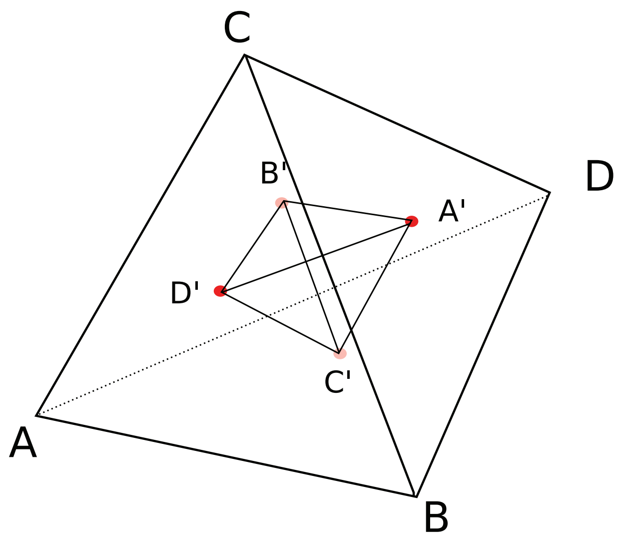
Whence

$$G^{-\frac{1}{2}}(A'-B')G^{-\frac{1}{2}} = E_{A'}-E_{B'} = \frac{1}{2}(E_B-E_A)+O(\epsilon^2) = \frac{1}{2}G^{-1}(B-A)+O(\epsilon^2)$$

Properties of the ALM mean

Recursively generalizable to $k \geq 4$ matrices $A_1, \ldots, A_k$

$$A_i^{(\nu+1)} = G(A_1^{(\nu)}, \ldots, A_{i-1}^{(\nu)}, A_{i+1}^{(\nu)}, \ldots, A_k^{(\nu)}), \quad i = 1, \ldots, k$$



- it satisfies the 10 ALM properties
- **problem**: slow convergence (linear with rate 1/2)
- **problem**: complexity $O(k! p^k n^3)$, p: number of iterations

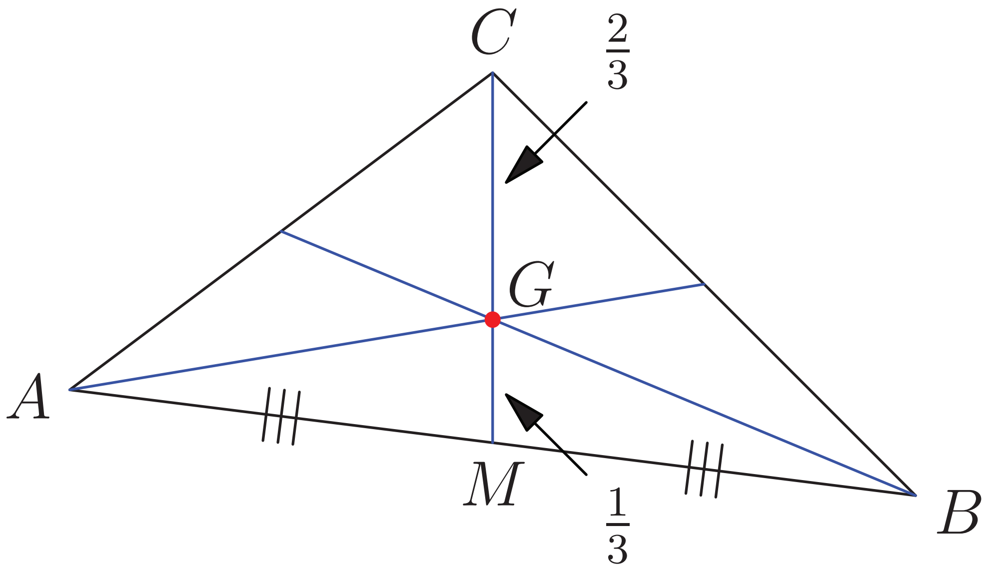
We may provide a different definition which leads to a substantial algorithmic improvement [Nakamura 2009], [B., Meini, Poloni, 2010]

In fact we overcome the first drawback about the slow convergence

It is based on the following

Remark

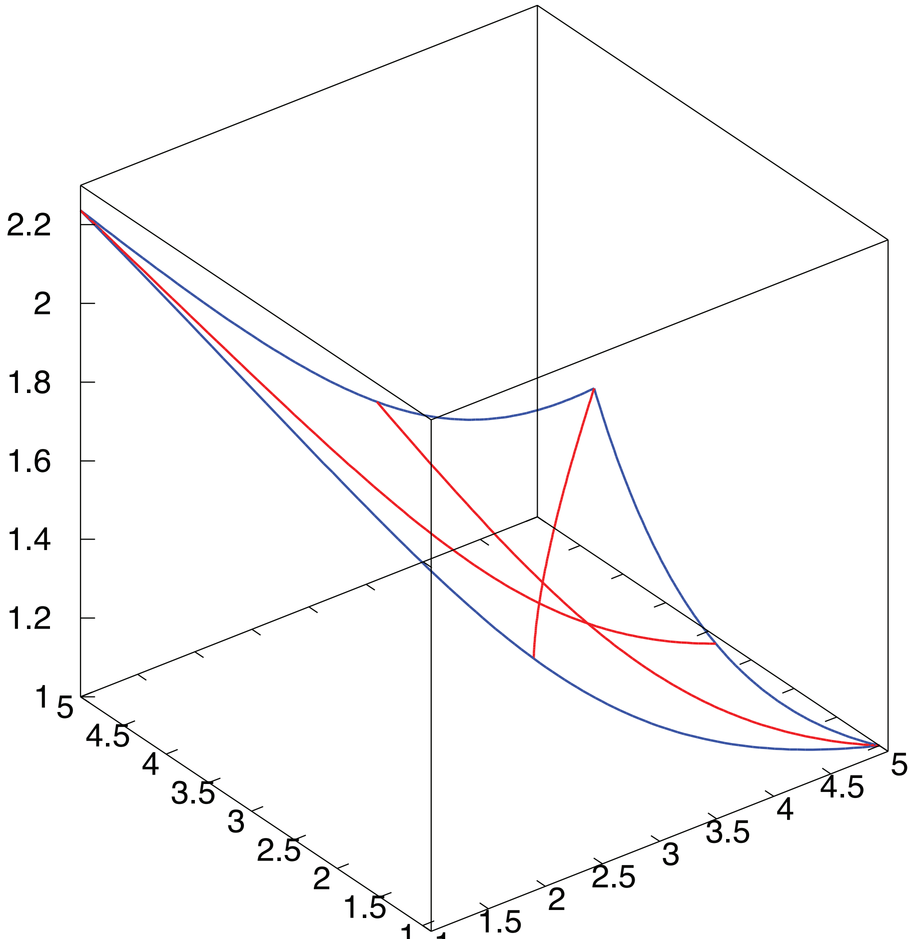
The three medians of a triangle meet at a single point, the centroid, at 2/3 of their length.



What happens with matrices?

Problem

In the Riemannian geometry the medians (geodesics) generally **do not intersect**



A new definition

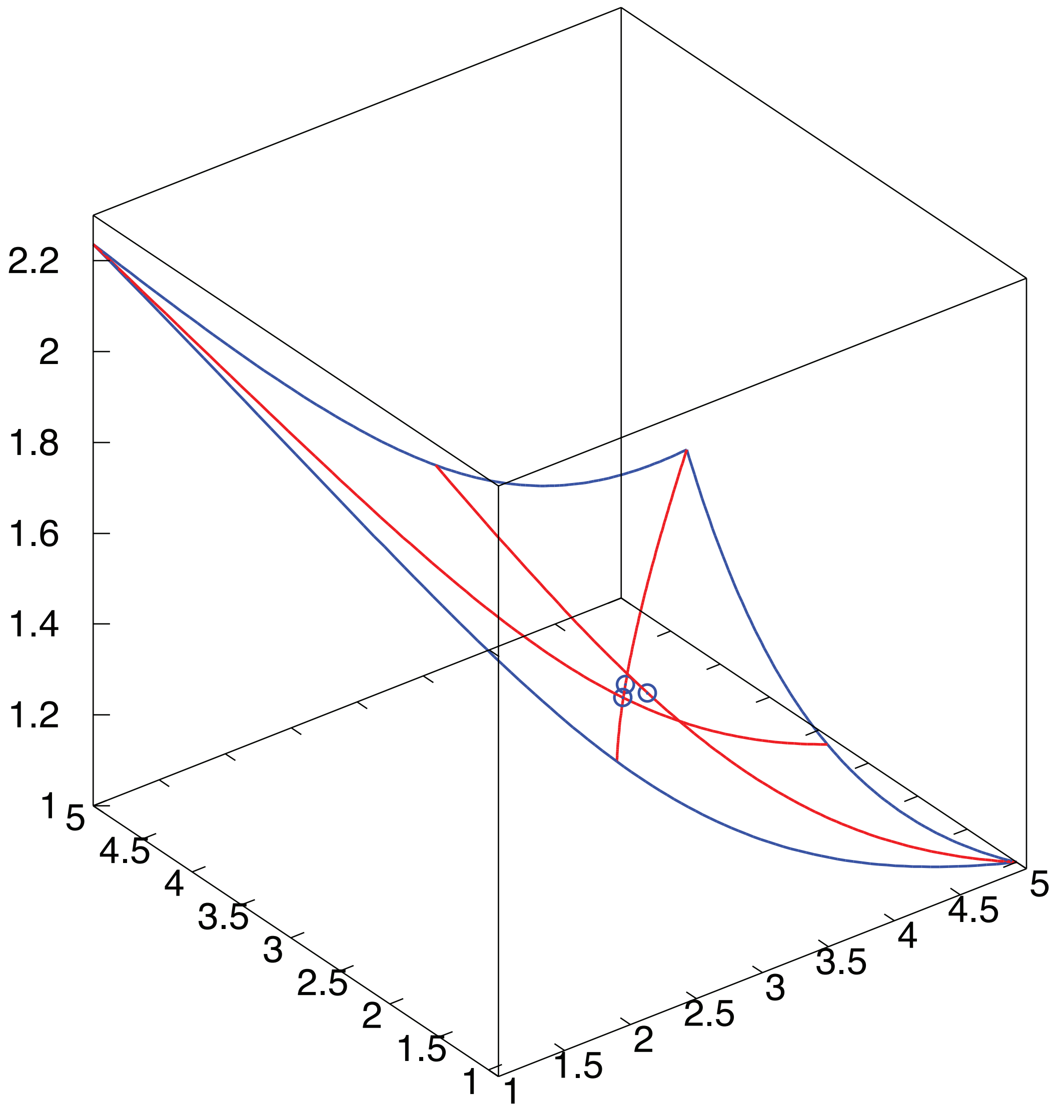
Define A_1, B_1, C_1 the points in the medians at distance $2/3$ from the vertices A, B, C, respectively.
Generally, A_1, B_1 and C_1 are pairwise different

Similarly, define A_2, B_2, C_2 from A_1, B_1, C_1 and so forth

$$A_{\nu+1} = A_\nu \#_{\frac{2}{3}} (B_\nu \#_{\frac{1}{2}} C_\nu)$$
$$B_{\nu+1} = B_\nu \#_{\frac{2}{3}} (C_\nu \#_{\frac{1}{2}} A_\nu)$$
$$C_{\nu+1} = C_\nu \#_{\frac{2}{3}} (A_\nu \#_{\frac{1}{2}} B_\nu)$$

where $A\#_t B := \gamma(t) = A(A^{-1}B)^t$

The new triangle is very small



Properties and remarks

Theorem

For any positive definite matrices A, B, C, the sequences A_ν, B_ν, C_ν obtained this way converge to the same limit. We define this limit $G(A, B, C)$ as the geometric mean of A, B, C

Theorem (Convergence with order 3)

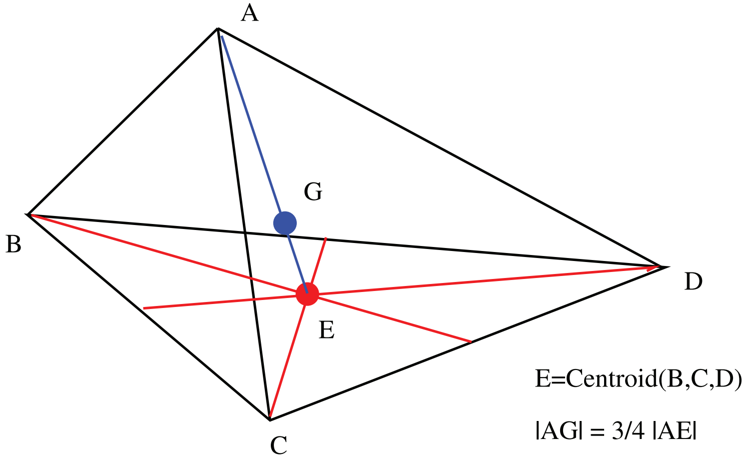
The convergence speed is of **order three**, *that is the error is $O(\lambda^{3^\nu})$, for $0 < \lambda < 1$, while for the ALM mean the error is $O(2^{-\nu})$.*

Theorem (ALM properties)

In general $G(A, B, C)$ is **different** *from $G_{ALM}(A, B, C)$, but fulfills the 10 ALM properties*

Properties and remarks

The mean can be generalized to $k \geq 4$ matrices; for $k = 4$ the barycenter of a tetrahedron is in the segment joining each vertex with the centroid of the triangle (facet) formed by the remaining points, at distance $3/4$; the nice properties are preserved.



In general one has:

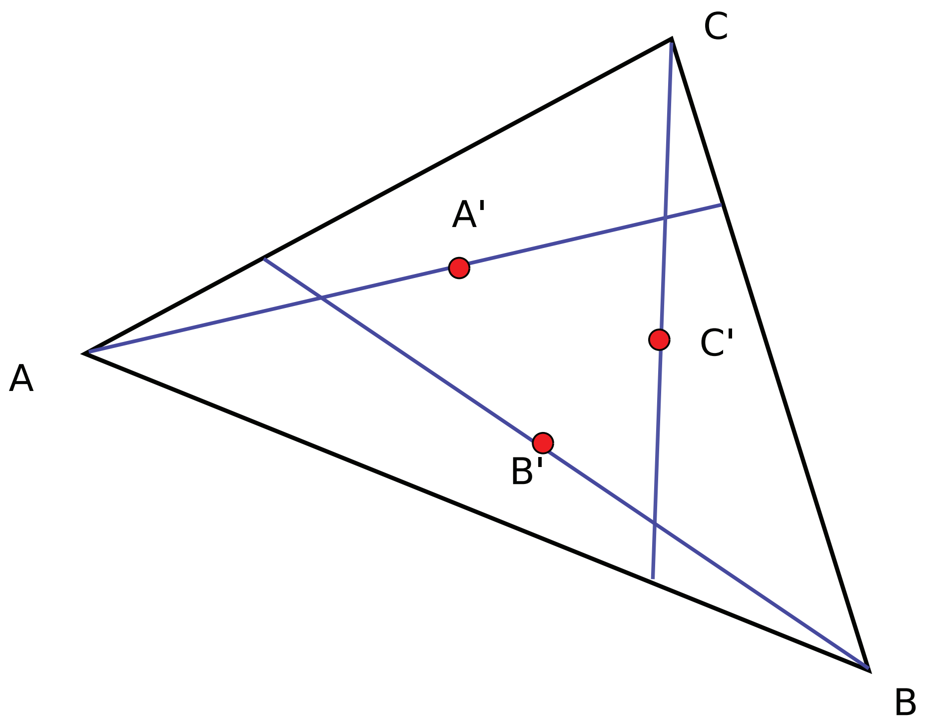
$$A_i^{(\nu+1)} = A_i^{(\nu)} \#_{\frac{k-1}{k}} G(A_1^{(\nu)}, \ldots, A_{i-1}^{(\nu)}, A_{i+1}^{(\nu)}, \ldots, A_k^{(\nu)}),$$

complexity $O(k!p^k n^3)$, p: number of iterations

Convergence of a general class of means

Given $0 < s, t \leq 1$ define the sequences

$$A' = A\#_s(B\#_t C), \quad B' = B\#_s(C\#_t A), \quad C' = C\#_s(A\#_t B)$$



Observe that for $s = 1$, $t = 1/2$ this gives the ALM mean.
For $s = 2/3$, $t = 1/2$ this gives the mean based on medians

Local convergence analysis

For $A, B, C, S \in \mathcal{P}_n$ it holds that

$$S(A\#_s(B\#_t C))S = (SAS)\#_s((SBS)\#_t(SCS))$$

Therefore, with the same notation as before, applying the above arguments to the matrix $A' = A\#_s(B\#_t C)$ yields

$$E_{A'} =(1-s)E_A + s(1-t)E_B + stE_C + \frac{st(t-1)}{2}H_{BC}^2 + \frac{s(s-1)}{2}(H_{BC} + tH_{AC})^2 + O(\epsilon^3)$$

where

$$H_{XY} = E_X - E_Y = G^{-\frac{1}{2}}(X - Y)G^{-\frac{1}{2}}$$

Whence

$$H_{A'B'} = (1-s)H_{AB} + s(1-t)H_{BC} + stH_{CA} + O(\epsilon^2)$$

In matrix form

$$\begin{bmatrix} H_{A'B'} \\ H_{B'C'} \\ H_{C'A'} \end{bmatrix} = \begin{bmatrix} (1-s)I & s(1-t)I & stI \\ stI & (1-s)I & s(1-t)I \\ s(1-t) & stI & (1-s)I \end{bmatrix} \begin{bmatrix} H_{AB} \\ H_{BC} \\ H_{CA} \end{bmatrix} + O(\epsilon^2)$$

The above matrix can be written in terms of the Kronecker product $\otimes$ as

$$\mathcal{C}(s,t) = \begin{bmatrix} 1-s & s(1-t) & st \\ st & 1-s & s(1-t) \\ s(1-t) & st & 1-s \end{bmatrix} \otimes I =: V(s,t) \otimes I$$

The matrix $V(s,t)$ is circulant, i.e., it is a polynomial in $U = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

$$V(s,t) = (1-s)I + stU + s(1-t)U^2$$

Moreover $\mathrm{spect}(U) = \{1, \omega, \bar{\omega}\}$, $\omega = -\frac{1}{2} + i\frac{\sqrt{3}}{2}$ with eigenvectors

$$w_1 = (1,1,1)^T, \quad w_2(1,\omega,\bar{\omega})^T, \quad w_3 = (1,\bar{\omega},\omega)^T$$

This way, we can write the eigenvalues of $V(s,t)$

$$\lambda_1 = 1, \quad \lambda_2 = 1 - \frac{3}{2}s + i\frac{\sqrt{3}}{2}s(2t-1), \quad \lambda_3 = \bar{\lambda}_2$$

and the corresponding eigenvectors are $w_1, \; w_2, \; w_3$

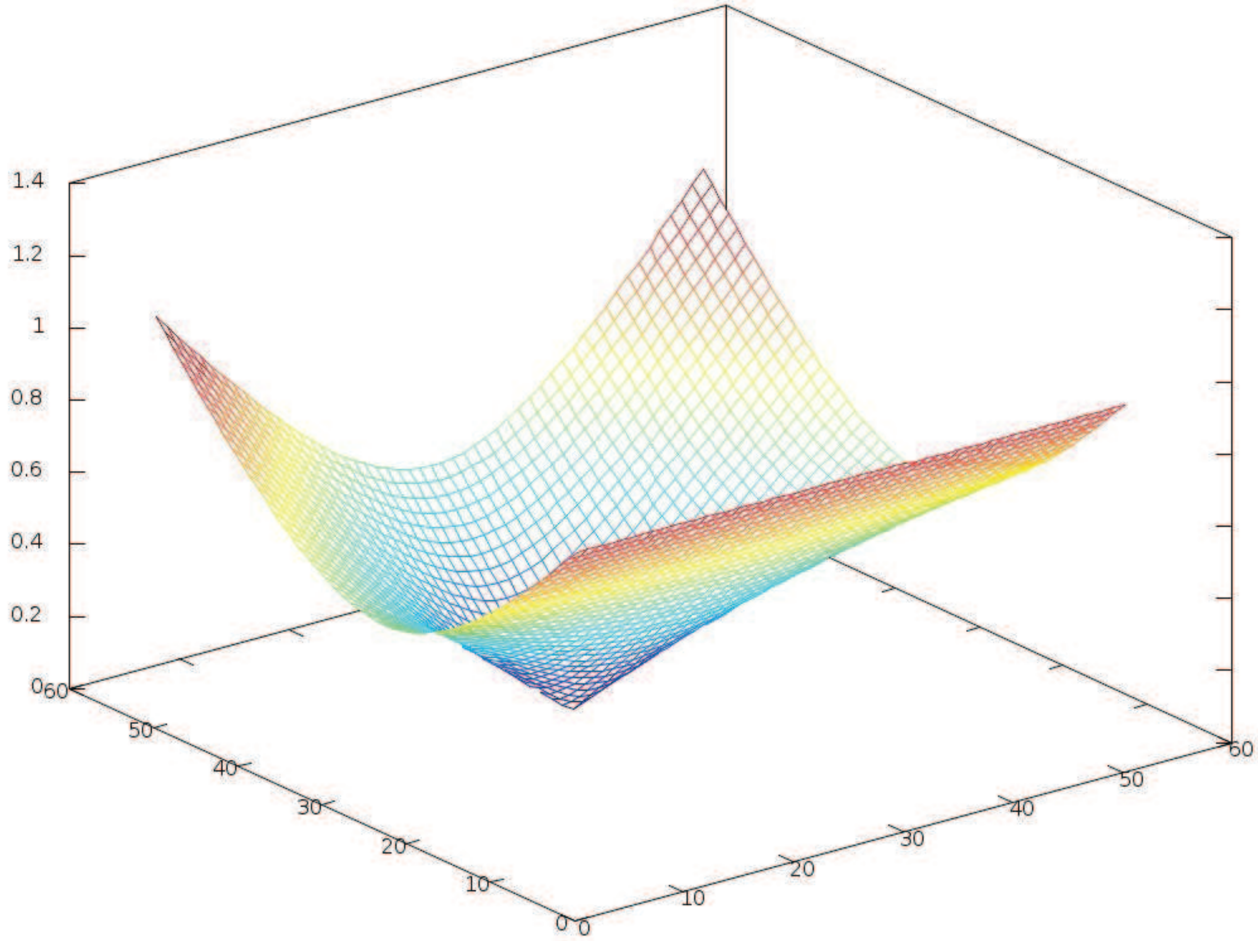
Therefore, from the properties of the Kronecker product, the matrix $\mathcal{C}(s,t)$ has eigenvalues $\lambda_1, \lambda_2, \lambda_3$ with multiplicity n and corresponding eigenvectors $w_j \otimes y$, $j = 1, 2, 3$, where $y \in \mathbb{R}^n$ is any nonzero vector.

Since $H_{AB} + H_{BC} + H_{CA} = 0$, the columns of the matrix $\begin{bmatrix} H_{AB} \\ H_{BC} \\ H_{CA} \end{bmatrix}$ are orthogonal to the eigenvectors corresponding to $\lambda_1 = 1$.
Thus, the contraction factor is bounded by

$$\varphi = |\lambda_2| = |\lambda_3| = \sqrt{(1 - \frac{3}{2}s)^2 + s^2(2t-1)^2}$$

For $s = 2/3$, $t = 1/2$ it holds that $\varphi = 0$.

Convergence rate



A closer analysis shows that for $s = 2/3$, $t = 1/2$ also the quadratic part of the error is zero. That is the method has order of convergence 3.

Some numbers

5
1.92542947898189
2.90969918536362
2.35774114351751
2.61639158463414
2.48316587472793
2.54876054375880
2.51571460655576
2.53217471946628
2.52392903948587
2.52804796243998
2.52598752310721
2.52701749813482
2.52650244948321
2.52675995852183
2.52663120018107
2.52669557839604
2.52666338904971
2.52667948366316
2.52667143634151

5
2.59890269690271
2.53027293208879
2.53025171828977
2.53025171828977

Example

$$A = \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 3 \\ 3 & 3 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$

The entry a_{11} is displayed at step i
Left: ALM mean
Right: New mean

Physical applications: speedup by a factor of 200 (mean of $k = 6$ matrices)

Some implementation: plotting the medians

The following code plots the medians of a triangle in the Riemannian geometry

```
function plot_medians(a,b,c)
% plot_medians(a,b,c) plot the medians of the triangle
% of vertices a,b,c  in the Riemannian geometry
% a,b,c: 2x2 positive definite matrices

hold off;
plottriangle(a,b,c);
hold on;
plot_geodesic(a,sharp(b,c,0.5));
plot_geodesic(b,sharp(a,c,0.5));
plot_geodesic(c,sharp(a,b,0.5));
hold off;
```

Some implementation: zooming the medians

The following code makes a zoom of the medians in a neighborhood of the points at distance 2/3

```
function zoom_medians(a,b,c,delta)
% zoom_medians(a,b,c,delta) plots the medians of the triangle
% of vertices  a,b,c in the riemannian geometry in a neighborhood
% of the points at distance 2/3
% a,b,c: 2x2 positive definite matrices
% delta: radius of the neighborhood
hold off;
bc=sharp(b,c,0.5); p1=sharp(a,bc,2/3-delta);
p2=sharp(a,bc,2/3+delta); plot_geodesic(p1,p2);   hold on;

ac=sharp(a,c,0.5); p1=sharp(b,ac,2/3-delta);
p2=sharp(b,ac,2/3+delta); plot_geodesic(p1,p2);

ab=sharp(a,b,0.5); p1=sharp(c,ab,2/3-delta);
p2=sharp(c,ab,2/3+delta); plot_geodesic(p1,p2);   hold off;
```

Some implementation: The ALM mean

```
function [ALM,iter]=alm(varargin)
% alm(A1,A2,...,Ak) computes the Ando-Li-Mathias mean of A1,...Ak
A=varargin;n=length(A{1}); k=length(A);  maxiter=200;  tol=1.d-14;
if k==2
   ALM=sharp(A{1},A{2},1/2);
else
   for iter=1:maxiter
      for h=1:k
         s=mod(h:h+k-2,k)+1;  A1{h}=alm(A{s});
      end
      ni=norm(A1{1}-A{1});
      if ni<tol   break   end
      A=A1;
   end
   ALM=A1{1};
   if iter==maxiter  disp('Max number of iterations reached')  end
end
```

Some implementation: the mean based on medians

```
function [G,iter]=nbmp(varargin)
% [G,iter]=nbmp(A1,A2,...,Ak) computes the NBMP mean
A=varargin; n=length(A{1}); k=length(A);  maxiter=200; tol=1.d-14;
if k==2
   G=sharp(A{1},A{2},1/2);
else
   for iter=1:maxiter
      for h=1:k
         s=mod(h:h+k-2,k)+1;
         A1{h}=sharp(A{h},nbmp(A{s}),(k-1)/k);
      end
      ni=norm(A1{1}-A{1});
      if ni<tol   break    end
      A=A1;
   end
   G=A{1};
   if iter==maxiter  disp('Max number of iterations reached')  end
end
```

Some implementation: the (s,t)- mean

```
function [G,iter]=st_mean(a,b,c,s,t)
% [G,iter]=st_mean(a,b,c,s,t) computes the (s,t)-mean of 3 matrices
n=length(a);  k=3;  maxiter=200;  tol=1.d-14;
for iter=1:maxiter
    a1=sharp(a,sharp(b,c,t),s);
    b1=sharp(b,sharp(c,a,t),s);
    c1=sharp(c,sharp(a,b,t),s);
    ni=norm(a1-a);
    if ni<tol   break   end
    a=a1;  b=b1;  c=c1;
end
G=(a1+b1+c1)/3;
if iter==maxiter  disp('Max number of iterations reached')   end
```

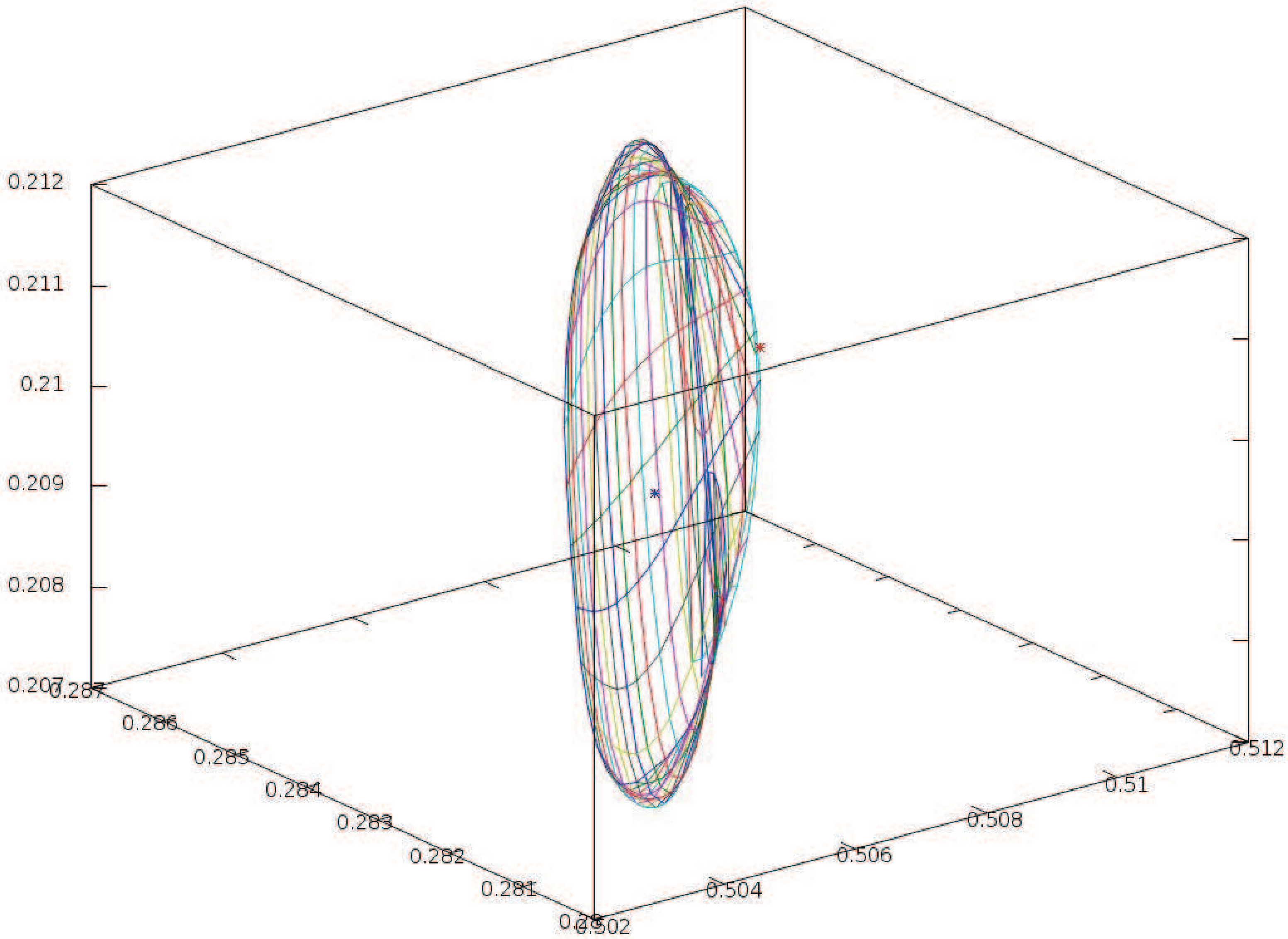
Some experiments

Plotting the (s,t)- mean for different values of s and t

```
function [x,y,z,iter]=plot_st_mean(a,b,c)
% plot_st_mean(a,b,c) makes a plot of the means of a,b,c
% obtained with different values of the parameters s and t

np=20;  t=0.1:1/(np-1):1;  s=0.1:1/(np-1):1;
n=length(t);
for i=1:n
   for j=1:n
      [g,it]=st_mean(a,b,c,s(i),t(j));
      x(j,i)=g(1,1); y(j,i)=g(1,2); z(j,i)=g(2,2);
      iter(i,j)=it;
   end
end
plot3(x,y,z);  hold on;  plot3(x',y',z');
galm=alm(a,b,c);  gnbmp=nbmp(a,b,c);
plot_matrix(galm,'r*');
plot_matrix(gnbmp,'b*');
hold off;
```

Some experiments



Some experiments: Computing distances

```
function [dr,de]=dist(A,B)
% [dr,de] = dist(A,B) computes the distance between A and B
%        in the Riemannian metric and in the Euclidean metric
%    dr: the Riemannian distance between A and B
%    de: the Euclidean distance between A and B

mA = cond(A); mB = cond(B);
if mA<=mB
    RA = chol(A); IRA = inv(RA);
    [U V] = schur(IRA'*B*IRA);  dr = norm(log(diag(V)));
else
    RB = chol(B); IRB = inv(RB);
    [U V] = schur(IRB'*A*IRB);  dr = norm(log(diag(V)));
end
de = norm(A-B,'fro');
```