עיבוד ספרתי של גיאומטריה – ש"ב 4

אייל טולצ'ינסקי 311505564, מרבי צ'יקוושוילי 317655157

תוכן עניינים

הסבר השיטה

פונקציית החום היא פונקציה $k_{t,x}(y)$ המודדת את החום בנקודה y אשר מקורו בנקודה x לאחר זמן t. ניתן לחשוב על כך כנגיעה עם מחט לוהטת בנקודה x ומעקב אחר התפשטות החום.

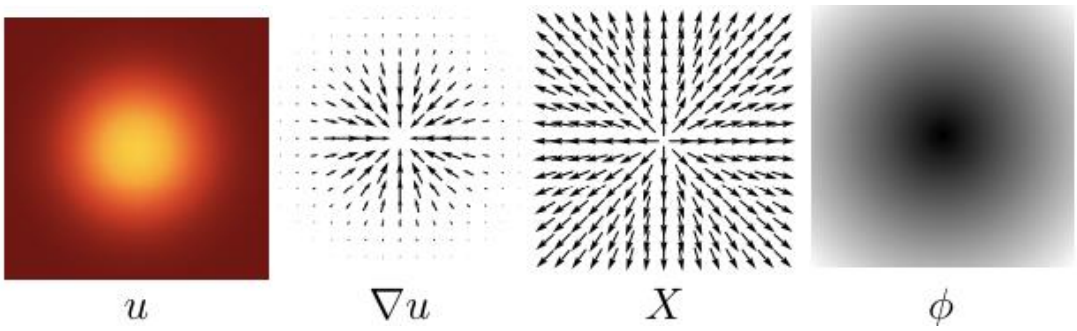
ורדאן הראה את הקשר בין פונקצית החום לפונקצית המרחק: $\varphi(x,y) = \lim_{t\to 0}\sqrt{-4t\log k_{t,x}(y)}$

כאשר $\varphi(x,y)$ הוא המרחק הגיאודזי בין שתי הנקודות.

לצערינו, שיטה זו רגישה מאוד לשגיאות נומריות, ולכן לא נשתמש בה בצורה זו.

ממשוואת אייקון אנו יודעים כי $|\nabla\varphi| = 1$, ולכן נוכל להשתמש בכך.
אנו נמצא פונקציה הדומה לפונקצית החום בכך שהגרדיאנטים שלהם מקבילים (על הפונקציה להיות רדיאלית ומונוטונית יורדת עם המרחק).
לאחר מכן נחשב את הגרדיאנט שלה ונהפוך אותו (תחילה מצביע בכיוון ההפוך, אל תוך המקור) וננרמל את הווקטורים, אותם נסמן X.
לבסוף נמצא את φ. לשם כך נצטרך לפתור משווה $\nabla\varphi = X$, או באופן שקול $\Delta\varphi = \nabla\cdot X$ (הכפלנו את שני הצדדים בדיברגנס).

נסמן את u להיות הפונקציה הרדיאלית היורדת עם המרחק. הקשר בין u ל-φ מתואר כאן:



האלגוריתם אם כן יהיה:

Algorithm 1 The Heat Method

I. Integrate the heat flow $\dot{u} = \Delta u$ for some fixed time t.
II. Evaluate the vector field $X = -\nabla u/|\nabla u|$.
III. Solve the Poisson equation $\Delta\phi = \nabla\cdot X$.

נותר רק להבין כיצד אנו מוצאים את u.

נסמן את u_0 (החום במרחב בזמן ההתחלתי), אשר יהיה שווה ל-1 בנקודות המקור, ו-0 בכל השאר (זוהי פונקצית דיראק).

משיטת אויילר, אנו יודעים כי בקירוב $\frac{u_t-u_0}{t} \approx \Delta u_t$, ולכן נוכל למצוא את u_t ע"י פתרון המשוואה הלינארית $(id - t\Delta)u_t = u_0$.
במחשבה ראשונה נעדיף לבחור את t להיות כמה שיותר קטן, אך למעשה ככל שהוא יותר קטן המרחק מתקרב למרחק הקומבינטורי – מספר הקשתות בין צומת המקור ליעד. כמו כן, ככל ש-t גדל המרחק שנמצא יהיה מעודן יותר, וגם זה לא בהכרח עדיף.

עפ"י הנאמר למעלה, את t נבחר להיות $t = h^2$ כאשר h אורך הקשת הממוצע (זוהי בחירה הנדסית אותה ביצעו כותבי המאמר על סמך ניסויים).

לסיכום, השיטה כולה היא פתירת משוואה לינארית (שיטת אויילר הפוכה), חישוב ונירמול הגרדיאנט של פונקציית החום, ופתירת משוואה לינארית נוספת, משוואת פואסון, לקבלת המרחק הסופי.

מימוש

כאמור, השיטה היא בסה"כ פשוטה ומסתמכת על אופרטורים דיפרנציאליים אותם כבר חישבנו בש"ב קודמים. סייג לכך הוא הלפלסיאן – אנו חישבו את ה-semi positive definite cotan weights בעוד בשיטה משתמשים ב-semi negative definite cotan weights, משמעות הדבר היא בסה"כ הפיכת הסימן.

נתאר את הפונקציות החדשות לש"ב אלו:

Mean_edge

מקבלת את הצמתים והפיאות, ומחזירה את אורך הצלע הממוצעת.

```
function [ h ] = mean_edge( verts,faces )
v1 = verts(faces(:,1),:);
v2 = verts(faces(:,2),:);
v3 = verts(faces(:,3),:);

e1 = v2-v1;
e2 = v3-v2;
e3 = v1-v3;

h = mean2([norm_row(e1), norm_row(e2), norm_row(e3)]);

end
```

Heat_func

מקבלת את הצמתים, הפיאות, צעד הזמן t ואת צומת/צמתי המקור gama, ומחזירה קירוב לפונקצית החום בזמן t.

היא יוצרת את הפונקציה u_0 להיות פונקצית דיראק המקבלת 1 על צמתי המקור ו-0 בשאר (כמתואר למעלה), ובעזרת צעד אויילר הפוך בעזרת הלפלסיאן, מוצאת קירוב ל-u_t.

```
function [ u ] = heat_func( verts,faces, t, gama, L )
if nargin < 5
    [~,~,L] = gen_diff_ops(verts, faces);
end
L = -L;

u0 = zeros(size(verts,1),1);
u0(gama) = 1;

u = (speye(size(L))-t*L) \ u0;

end
```

Dist_func

מקבלת את הצמתים, הפיאות ואת פונקצית החום., ומחזירה את פונקצית המרחק.
היא מחשבת את הגרדיאנט של פונקצית החום, הופכת את כיוונו ומנרמלת. לאחר מכן פותרת את משוואת פואסון ומוצאת את פונקצית המרחק.

למרחק זה נדרש תיקון, שכן היו בו איברים שליליים. לכן החסרנו מכל איבר את הערך המינימאלי של הפונקציה

```
function [ phi ] = dist_func( verts,faces,u, grad, div, L
)
%DIST_FUNC Summary of this function goes here
%   Detailed explanation goes here

if nargin < 6
    [grad,div,L] = gen_diff_ops(verts,faces);
end
L = -L;

X = -normalize_vectors(grad*u);
phi = L \ (div * X);
phi = phi - min(phi);

end
```

Find_border

מקבלת את הצמתים והפיאות, ומחזירה את האינדקסים לצמתים על הגבול.

לשם כך אנו יוצרים מטריצה דלילה, כאשר לכל קשת (i,j) נשים 1 בתא ה-(i,j). כיוון שהמטריצה אמורה להיות סימטרית (הקשתות אינן מכוונות) נוסיף לה את השחלוף שלה.
מדרך פעולתה של יצירת מטריצה דלילה, ומכך שהפכנו אותה לסימטרית, אם הקשת מופיעה מספר פעמים, הערך של התא המתאים יהיה כמספר הפיאות שהיא נמצאת בהן.

כעת נוציא את האינדקסים לכל השורות בהן מופיע האיבר 1. כפי שנאמר, אלו שורות i כך שקיימת קשת (i,j) המופיע רק בפיאה אחת. כלומר צמתים i שמהם יש קשת על הגבול, ומכאן שהצומת הוא על הגבול.

סה"כ הקוד הוא:

```
function [ border ] = find_border( verts, faces )
%FIND_BORDER Summary of this function goes here
%   Detailed explanation goes here

nverts = size(verts,1);
edges = sparse(faces, faces(:,[2 3 1]),1, nverts,
nverts);
edges = edges + edges';

[border,~] = find(edges == 1);
```

```
border = unique(border);


end
```

Find_point

הפעולה מקבלת את אוסף הצמתים וקוארדינטה, ומוצאת את הצומת הקרובה ביותר.

פעולה זו היא מאוד פשוטה – אנו מחשבים מרחק אוקלידי מכל נקודה את הנקודה שהתקבלה, ובוחרים את האחת הכי קרובה.

בעזרת פונקציה זו, מצאנו נקודה מעניינת על המש (ציירנו את המש וחיפשנו נקודה בעזרת העכבר), ולאחר מכן בקוד ניתן למצוא אותה בקלות.

```
function [ index ] = find_point( v,point )
%FIND_POINT Summary of this function goes here
%   Detailed explanation goes here

[~,index] = min(norm_row(v-repmat(point,...
                                 size(v,1),...
                                 1)));

end
```

Sphere_true_dist

הפונקציה מקבלת את הצמתים ואוסף צמתי מקור, ומחזירה לכל צומת את המרחק הגיאודזי המינימלי למקור.

בפונקציה זו השתמשנו כדי לחשב את המרחק הגיאודזי המדויק של נקודה על ספירה מאוסף צמתי מקור.

השתדלנו להימנע מלולאות, ולכתוב קוד יפה ומתוחכם יותר המשתמש במטריצות (כיאה במתלב) הרץ מהר יותר.

לשם כך, יצרנו מטריצה ארוכה בעלת 3 עמודות, המכילה שכפולים של צמתי המקור. כל צומת מקור מופיע $|V|$ פעמים ברציפות.

לאחר מכן, ע"י שכפול מטריצת הצמתים אנו מחשבים את המרחק בין כל נקודה לכל מקור (ע"י $(\cos^{-1}(P \cdot Q))$. מכך נקבל מטריצה ארוכה אשר $|V|$ האיברים הראשונים שלה הם המרחק מצומת המקור הראשון, הבאים מהצומת השני וכן הלאה.

Reshape של המטריצה הזו ל-$|V|$ שורות יביא לכך שבכל שורה i, נקבל עמודה לכל מקור j כך שבתא המתאים לצומת i (השורה ה-i) ולצומת המקור j מופיע המרחק ביניהם.

ע"י לקיחת המינימום בכל שורה אנו מוצאים את המרחק הגיאודזי המינימאלי של כל צומת למקור.

```
function [ true_dist ] = sphere_true_dist(v, gama )
points = reshape(v(gama,:),1,[]);
points = repmat(points,size(v,1),1);
points = reshape(points,[],3);
```

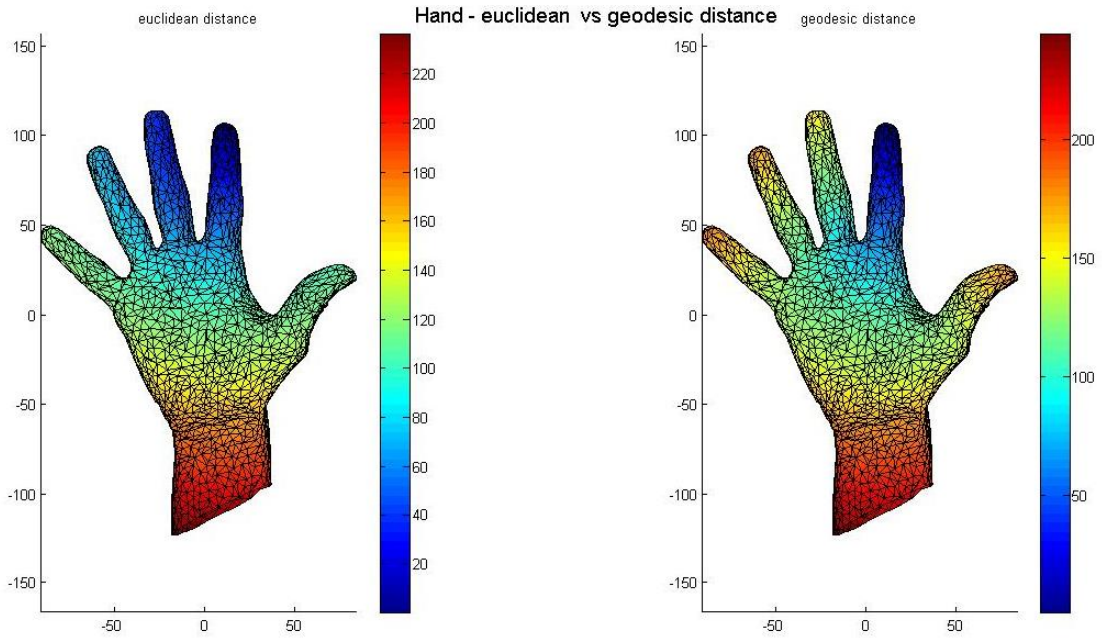
```
true_dist = dot(repmat(v,size(gama,2),1),points,2);
true_dist = acos(true_dist);
true_dist = reshape(true_dist, size(v,1),[]);
true_dist = min(true_dist,[],2);

end
```

דוגמות

מרחק אוקלידס מול מרחק לגיאודזי

במודל היד, בחרנו את המקור להיות קודקוד האצבע המורה. כפי שניתן לראות, על פי המרחק האוקלידי קודקוד האצבע האמה קרוב אליה, בעוד המרחק הגיאודזי אליו למעשה רחוק.

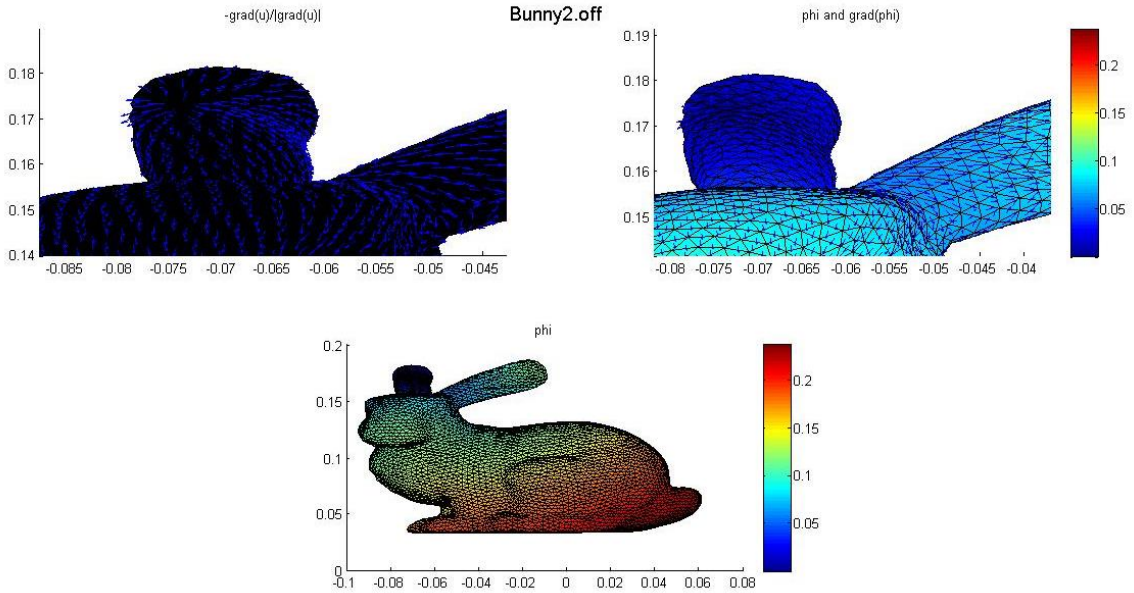


גרדיאנטים לפונקציות החום והמרחק

כאן בחרנו את המקור להיות האוזן של הארנב.

ניתן לראות כי אכן הגרדיאנט של פונקציית החום, וכן הגרדיאנט של פונקציית המרחק, משיקים למשטח, ומכוונים רדיאלית הרחק מהמקור.

כמו כן ניתן לראות בפונקצית המרחק כי האוזן השניה, שכביכול קרובה בקו אווירה, נמצאת רחוק על פי מרחק גיאודזי.



שגיאה במרחק - נקודה בודדת על ספירה

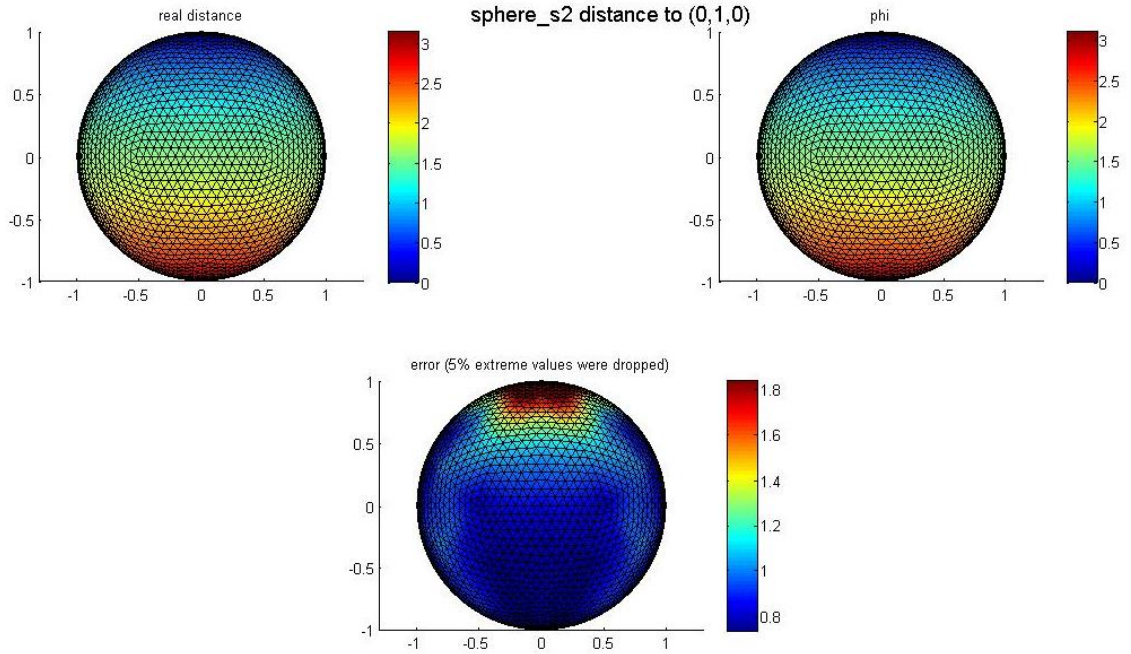
בחרנו את המקור להיות הנקודה העליונה בציור $(0,1,0)$.

המרחק המדויק בין נקודות על ספירה נתון כ-$\varphi(P,Q) = \cos^{-1}(P \cdot Q)$.

ניתן לראות כי המרחק שאנו חישבנו קרוב מאוד למרחק המדויק.

נציין כי בתמונה המציגה את השגיאה, הצבעים הותאמו כדי להתעלם מ-5% הערכים הגבוהים ו-5% הערכים הנמוכים, ולמעשה השגיאה בכל נקודה קטנה מאוד, כאשר עיקרה נמצא קרוב למקור.

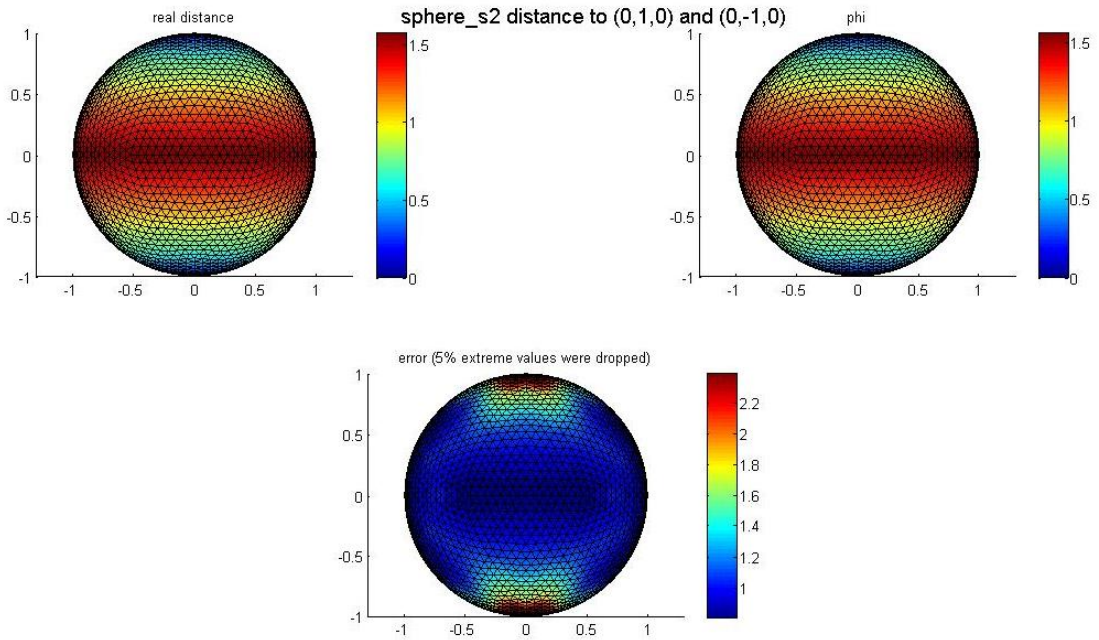
ייתכן והשגיאה קרוב למקור נובעת משגיאות נומריות (כפי שנראה מאוחר יותר, גם השגיאה המקסימלית גדלה כאשר אורך הקשת הממוצע קטן, דבר הנובע משגיאות נומריות כנראה).



שגיאה במרחק - זוג נקודות על ספירה

כעת בחרנו את המקור להיות זוג נקודות בקצוות הספירה - $(0,1,0), (0,-1,0)$.

ניתן לראות כי גם כאשר המקור אינו נקודה בודדת אלא זוג נקודות, עדיין אנו מקבלים תוצאות טובות, כאשר השגיאה העיקרית קרובה למקור.
שוב, בתמונת השגיאה מתעלמים מ-5% הערכים הקיצוניים, וככל הנראה קרוב למקור היא נובעת משגיאות נומריות.



מודלים עם גבול

מצאנו (לאחר מאמצים רבים) זוג מודלים עם גבולות. הראשונה היא המסכה, והשניה היא היד.

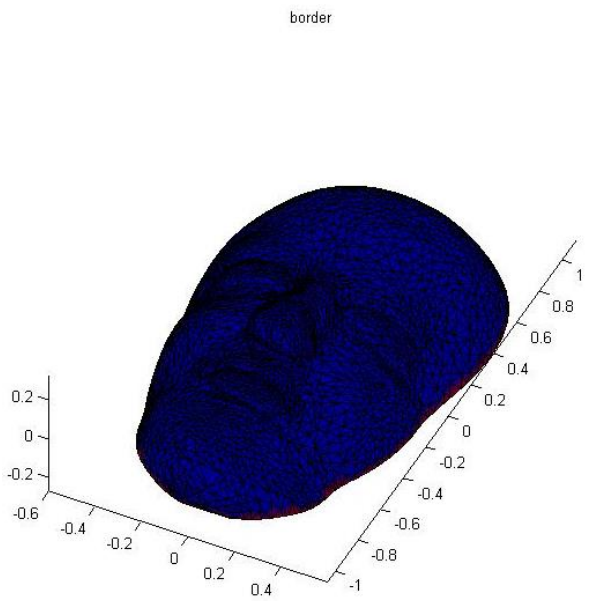
חשוב לאמר כי ביד, מעבר לגבול הברור בבסיס היד, יש עוד זוג גבולות באצבע המורה ובזרת (אשר התגלו לאחר התבוננות בתמונת המרחקים מהגבולות, אשר לא הגיוניות אם הגבולות הנ"ל לא קיימים).

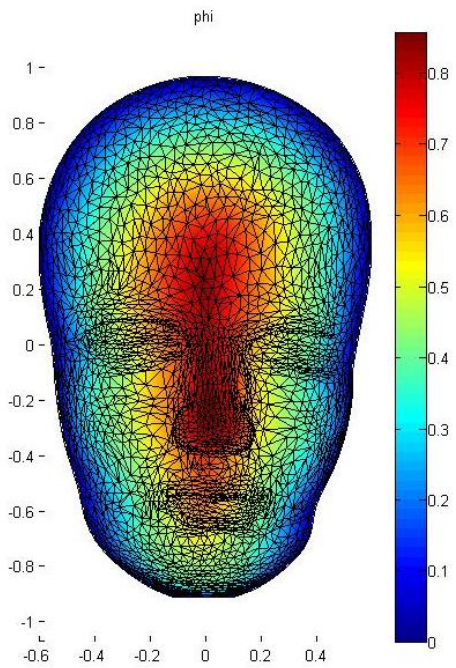
ניתן לראות כי גם כאן המרחק נראה למראית עין נכון. לצערנו במודלים המורכבים (שאינם הספירה), אין לנו נוסחה למציאת מרחק מדויק, ולכן לא יכולנו להשוות את התוצאות למרחק האמיתי ונאלצנו להסתפק בהתבוננות ובדיקה שהמרחק נראה הגיוני.

בצד שמאל מתוארת פונקצית דיראק אשר ערכה 1 על צמתי הגבול ו-0 בשאר. מטרת התמונה להציג את הגבול. מצד שמאל המרחק אל הגבול.

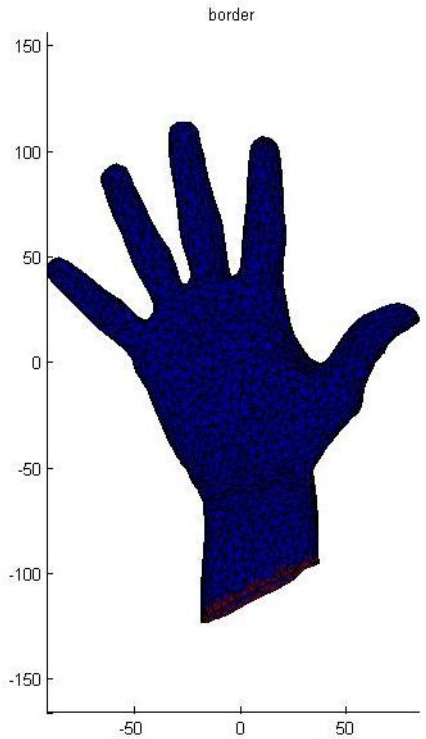
יפה גם לראות כי במודל המסכה (איזור האף), וכן במודל היד (מרכז היד, האגודל), ישנן נקודות אשר קרובות ביותר ליותר משתי נקודות במקור, ובכל זאת האלגוריתם נראה שעובד נכון.

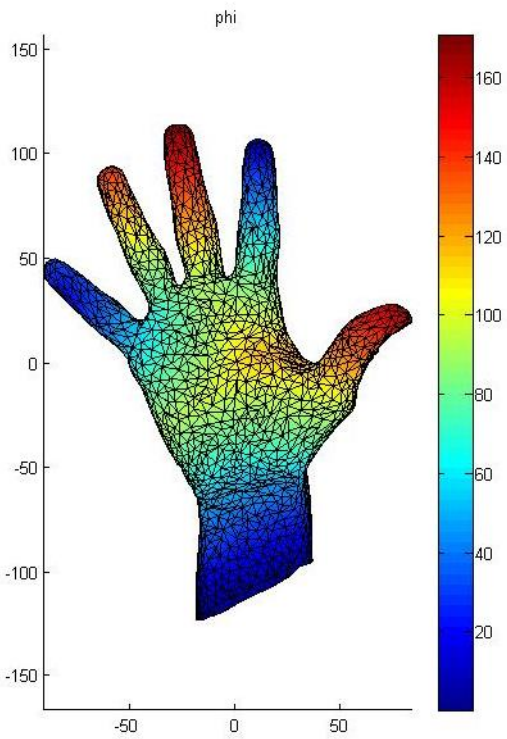
mask





hand





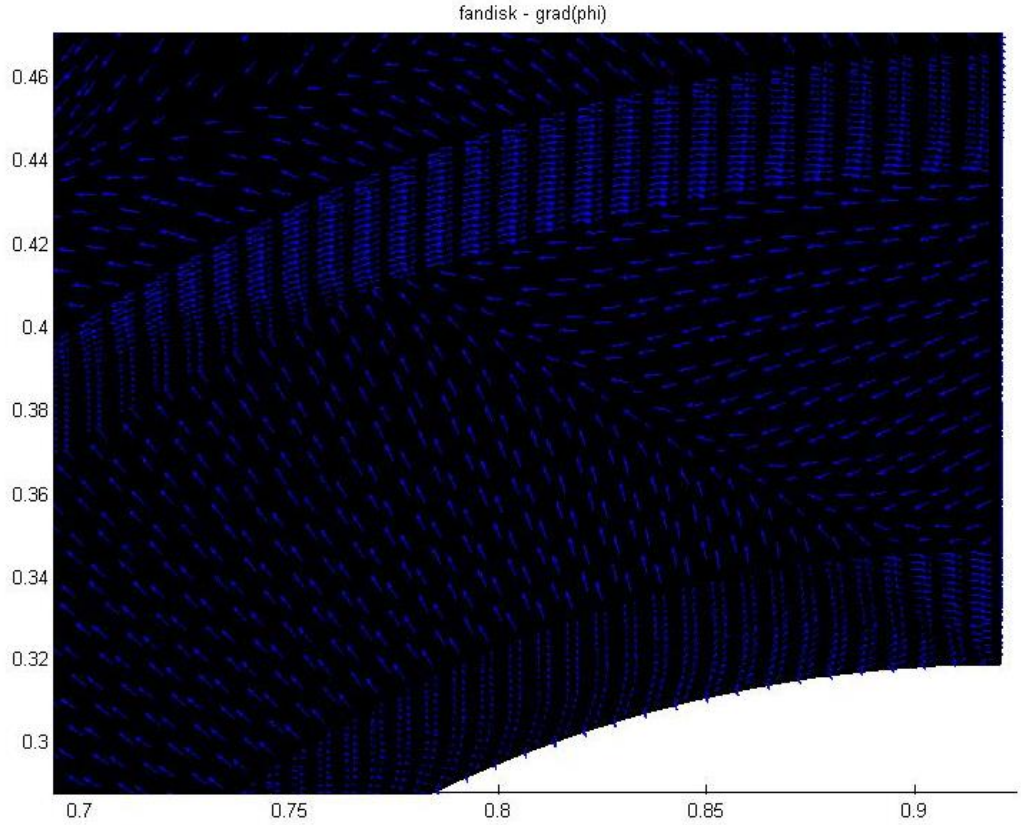
מודל עם פינות חדות

נציג את תוצאת האלגוריתם על מודל ה-fandisk. מודל זה בעל פינות חדות רבות, אשר מקשות לרוב על אלגוריתמים למרחקים גיאודזיים, ועל אלגוריתמים המשתמשים באופרטורים דיפרנציאליים ככלל.

ניתן לראות כי בכל זאת הגרדיאנט של פונקצית המרחק משיק למשטח (כצפוי) ומכוון בכיוון הנכון. כמו כן המרחק עצמו נראה נכון, ומקרב טוב את המרחק הגיאודזי האמיתי, על אף הפינות החדות.

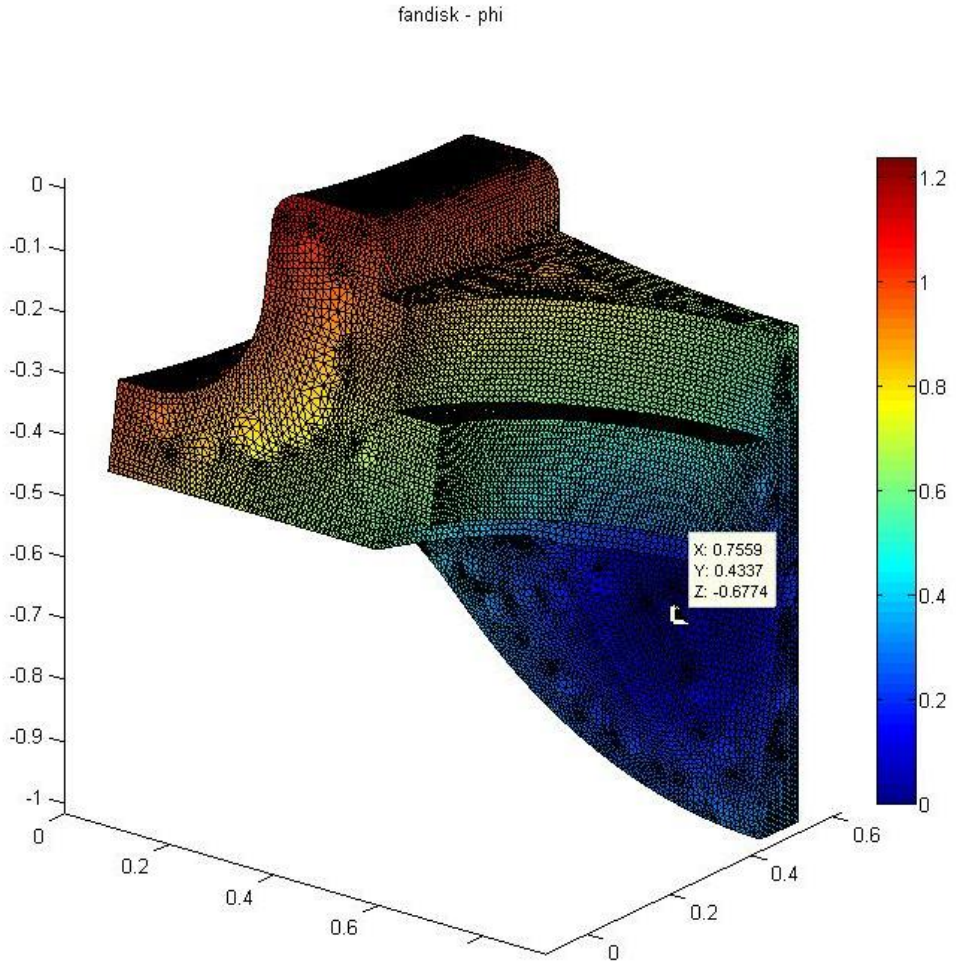
הגרדיאנט

כפי שנאמר, הגרדיאנט משיק למודל ומכוון נכון.



פונקצית המרחק

נקודת המקור מסומנת בתמונה, וניתן לראות את הפינות החדות אשר מקשות על האופרטורים הדיפרנציאליים. בכל זאת המרחק המתקבל מקרב לא רע את האמת.



שגיאה כפונקציה של t, וכפונקציה של דיוק המודל

כפי שנאמר קודם, בחירה של t קטן מדי לא בהכרח תקרב את המרחק.

אנו מציגים את אחוז השגיאה הממוצע והמקסימלי עבור מספר משים לבחירות שונות של m, כאשר $t = m \cdot h^2$ (h אורך הצומת הממוצע).
אחוז השגיאה – האחוז של השגיאה במרחק בכל צומת, מתוך המרחק האמיתי אל הצומת.

המודלים שבחרנו הם מודלים של רמות דיוק שונות לספירה:

Sphere_s{i} כאשר i נע בין 0 ל-4.

ניתן לראות מתוך הגרפים גם את השגיאה הממוצעת והמקסימלית כתלות בדיוק המודל, או באורך הקשת הממוצע (שפרופורציונאלי לדיוק המודל).

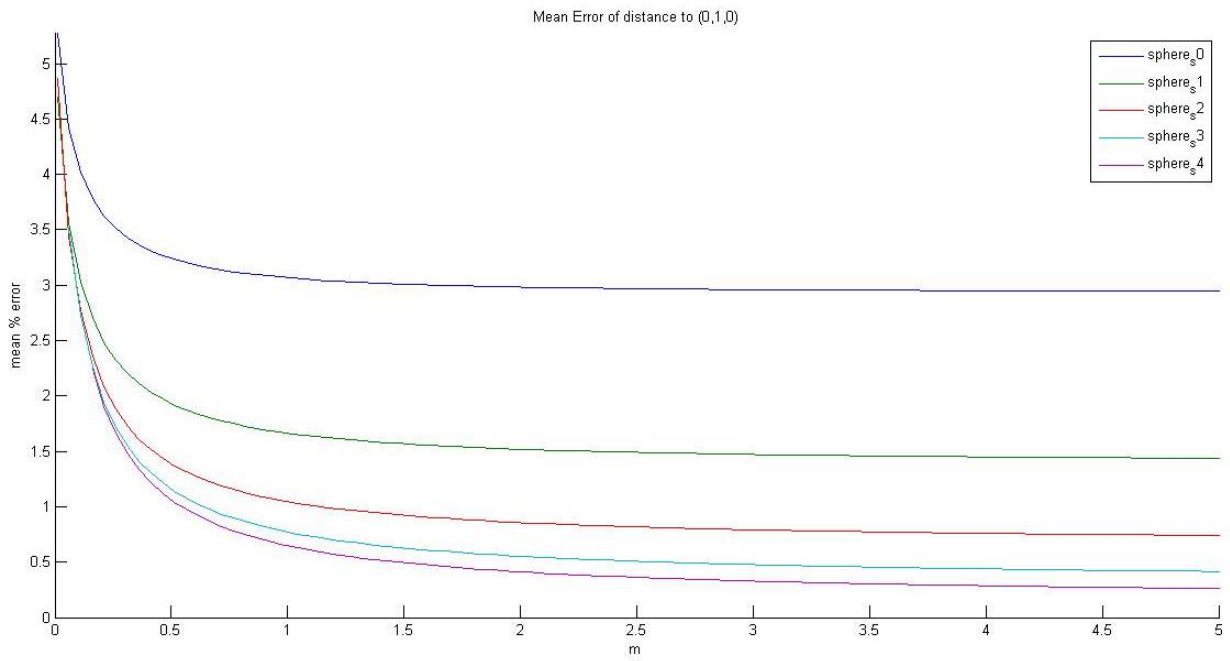
את ההשוואה נוכל לבצע אם נסתכל על השגיאה בכל אחד מהמודלים עבור ערך m זהה – זוהי בחירה יותר הוגנת מבחירת t זהה, שכן t תלוי כבר באורך הקשת הממוצע, אשר משתנה דרסטית בין המודלים.

ניתן לראות כי ככל שהדיוק הגדול עולה, ואורך הקשת הממוצע יורד, גם השגיאה הממוצעת קטנה.

השגיאה המקסימלית גם היא קטנה, עד שמגיעים לאורכי קשתות כה קטנים שבהם המרחקים רגישים לשגיאות נומריות. כתוצאה מכך ייתכן ומקבלים שגיאה (באחוזים) גבוהה באוסף קטן של צמתים, אשר אחראים לשגיאה המקסימלית.

התוצאות שלנו תואמות את המאמר, הטוען כי הבחירה המיטבית של m היא באיזור הערך 1.
במאמר מתואר הגרף עבור גרפים שונים, אשר לא בהכרח נוחים וקלים כמו הספירה (לא היה ביכולתנו לחשב את המרחק המדויק למודלים מורכבים), אשר בהם השגיאה גם גדלה עבור $m > 1$.

שגיאה ממוצעת



שגיאה מקסימלית