

## הדפסת מידע תלת מימדי

נבדוק לדוגמה איך להציג מטריצה שבה הערך בכל איבר הוא סכום האינדקסים שלו (מתקבל מישור משופע).

נראה שתי אפשרויות להצגת פלט חיצוני בעזרת gnuplot:

### pm3d .1

התכנית הבאה מבצעת הדפסה לקובץ חיצוני (בקטע המודגש).

התכנית פשוט מדפיסה את ערכי המטריצה אחד אחרי השני, כאשר חשוב לזכור כי בין כל שני ערכים באותה שורה מדפיסים רווח, ובין כל שתי שורות מדפיסים "\n".

```
#include<stdio.h>
#include <stdlib.h>

#define ROWS 20
#define COLUMNS 20
#define DX 10
#define DY 10

void **alloc2d(int, int, int);
void free2d(void **, int, int);

int main()
{
    FILE *out;
    int i,j;
    float **mat;

    mat=(float**) alloc2d(ROWS,COLUMNS,sizeof(float));
    out=fopen("matrix_for_3D_pm3d","w");

    for (i=0;i<ROWS;++i) {
        for (j=0;j<COLUMNS;++j) {
            mat[i][j]=i+j;
            fprintf(out,"%f ", mat[i][j]);
        }
        fprintf(out,"\n");
    }
}
```

```

    free2d((void **)mat,ROWS,COLUMNS);
    fclose(out);

    return 0;
}

void **alloc2d( int d1, int d2, int size) {
    int i,j;
    void **p;
    p = (void**)malloc(d1*sizeof(void*));

    if ( p == NULL ) return NULL;
    for ( i=0; i<d1; ++i ) {
        p[i] = malloc(d2*size);
        if ( p[i] == NULL ) return NULL;
    }
    return p;
}

void free2d( void **p, int d1, int d2 ) {
    int i,j;

    if ( p != NULL ) {
        for ( i=0; i<d1; ++i ) {
            if ( p[i] != NULL )
                free(p[i]);
        }
        free(p);
    }
}

```

לאחר שמקמפלים ומריצים את התכנית, נותר להדפיס את קובץ הפלט בעזרת `gnuplot`. לשם כך נכנסים ל-`gnuplot` ורושמים:

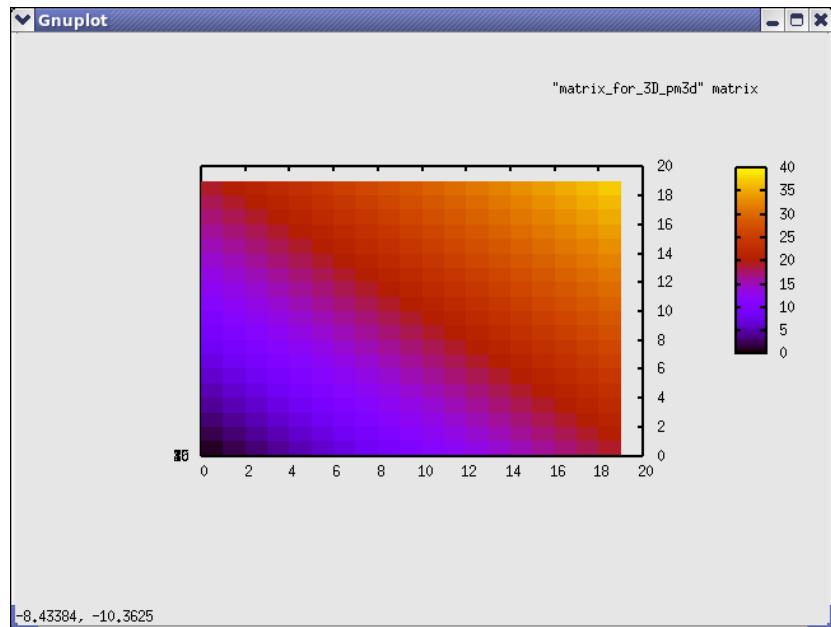
```

set pm3d
splot "matrix_for_3D_pm3d" matrix with pm3d

```

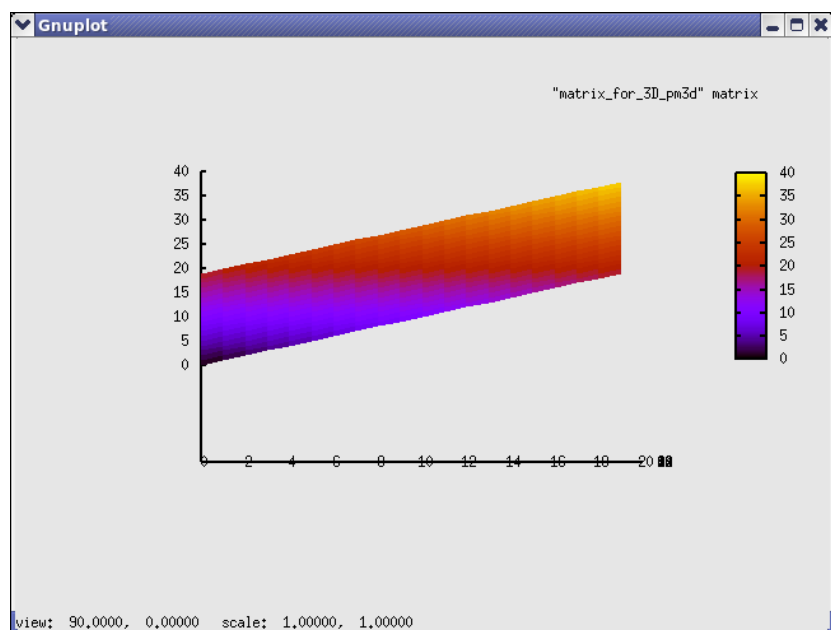
כתוב שם "matrix\_for\_3D\_pm3d" כי זה שם הקובץ שנוצר בתכנית שלמעלה. כמוכן אפשר לבחור איזה שם שרוצים עבור הקובץ שבו יהיה הפלט.

כך ה-gnuplot יידע לזהות את המטריצה שהוא אמור לצייר, ומקבלים:



זהו מבט על. כפי שרואים כאן, המטריצה היא בגודל 20X20 וערכי איבריה נעים בין 0 ל-40.

לאחר ש-gnuplot מוציא את התמונה הזו, ניתן לסובב אותה בעזרת הכפתור השמאלי בעכבר, ולהגיע למבט מזווית שונה, למשל מבט צד כזה:



בתמונה שמעל, לפי הרישום בצד שמאל למטה רואים מאיזו נקודת מבט נלקחה התמונה.  
אפשר גם לקבוע זאת בעזרת כתיבת הפקודה 'set view' לפני השימוש ב-`splot`.  
לדוגמה, כדי לראות במבט על, רושמים:

```
set view 0,0
```

וכדי לראות במבט צד, אפשר לרשום:

```
set view 90,0
```

ואז להציג את התוצאה עם הפקודה `splot` כפי שהוזכר למעלה.

## 2. dgrid3d

בדרך השנייה שאפשר לבחור, מדפיסים בכל פעם 3 דברים: קואורדינאטה x של השדה, קואורדינאטה y שלו (או z לצורך העניין) וערך של מטריצת השדה בנקודה זו. ניתן לראות זאת בקטע המודגש.

```
#include<stdio.h>
#include <stdlib.h>

#define ROWS 20
#define COLUMNS 20
#define DX 10
#define DY 10

void **alloc2d(int, int, int);
void free2d(void **, int, int);
int main()
{
    FILE *out;
    int i,j;
    float **mat;
    mat=(float**) alloc2d(ROWS,COLUMNS,sizeof(float));
    out=fopen("matrix_for_3D","w");

    for (i=0;i<ROWS;++i) {
        for (j=0;j<COLUMNS;++j) {
            mat[i][j]=i+j;
            fprintf(out,"%d %d %f\n", i*DX, j*DY, mat[i][j]);
        }
    }
}
```

```

    free2d((void **)mat,ROWS,COLUMNS);
    fclose(out);

    return 0;
}

void **alloc2d( int d1, int d2, int size) {
    int i,j;
    void **p;
    p = (void**)malloc(d1*sizeof(void*));

    if ( p == NULL ) return NULL;
    for ( i=0; i<d1; ++i ) {
        p[i] = malloc(d2*size);
        if ( p[i] == NULL ) return NULL;
    }
    return p;
}

void free2d( void **p, int d1, int d2 ) {
    int i,j;

    if ( p != NULL ) {
        for ( i=0; i<d1; ++i ) {
            if ( p[i] != NULL )
                free(p[i]);
        }
        free(p);
    }
}

```

לאחר קומפילציה והרצה, נכנסים ל- gnuplot ורושמים:

```

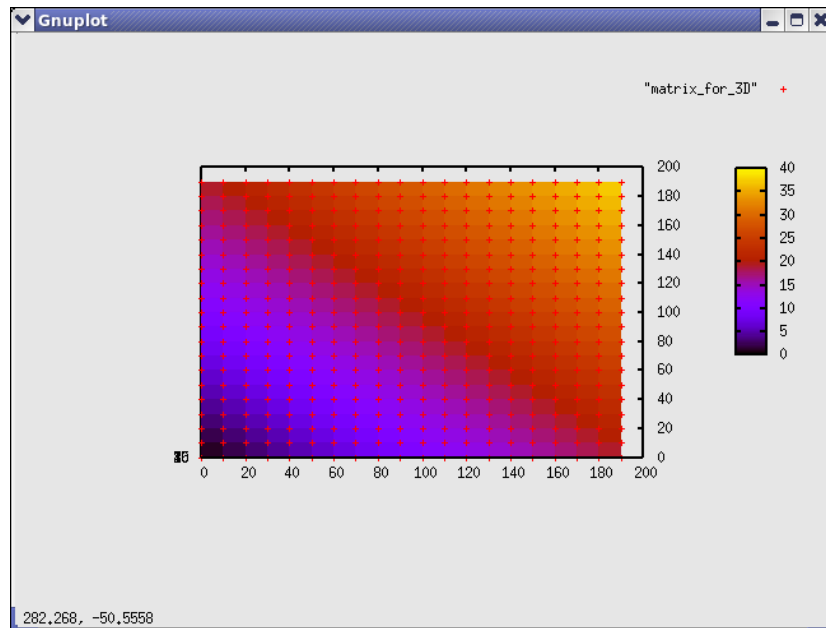
set dgrid3d 20,20
splot "matrix_for_3D"

```

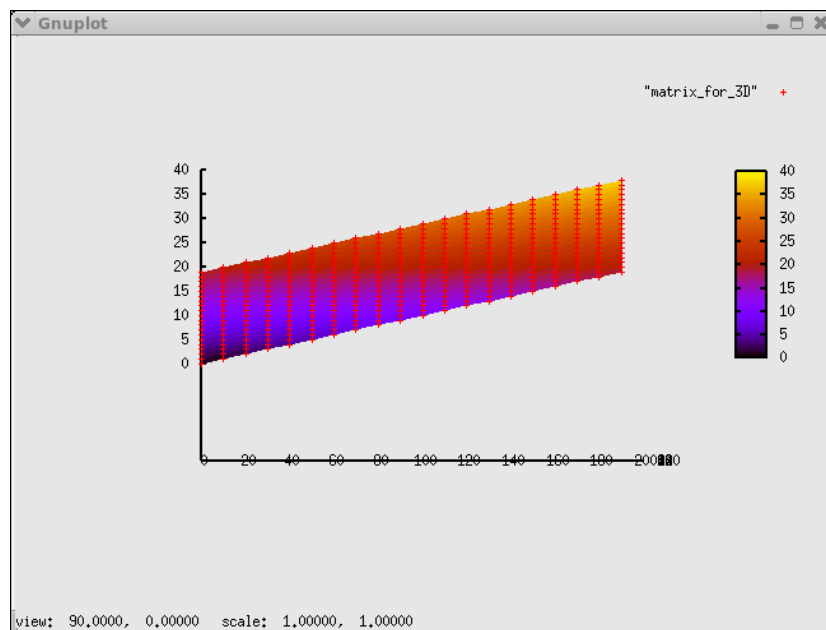
המספרים 20, 20 מתייחסים למימדי המטריצה בדוגמה. זה כמובן משתנה בהתאם לגודל המטריצה שרוצים לצייר.

שוב, "matrix\_for\_3D" זה שם הקובץ שניתן בדוגמה הזו, אך כמובן אפשר לבחור מה שרוצים.

כך נראית התמונה במבט על:



וכך התמונה נראית במבט צד:



היתרון של הצגת pm3d:

עיבוד מידע מהיר יותר – טוב למערכים גדולים.

היתרון של הצגת dgrid3d:

התמונה מציגה את הקואורדינאטות האמיתיות של x ו-y, בניגוד לשיטת pm3d שמציגה פשוט

את האינקדסים של המטריצה. ניתן לראות את ההבדל הזה בדוגמאות שהובאו למעלה.

בתרגיל הסיום – מציגים את שדה הגלים ואת שדה החום בזמנים מסוימים לפי אחת השיטות שראינו כאן.