

Lecture 10
CS 137
Fall 2014
by Chantelle Gellert

Fixed point iteration

root x_0 such that $f(x_0) = 0$, using fixed point, x such that $g(x_1) = x_1$

$g(x) = x$ has infinite fixed points

convert one to the other

$f(x) = x^3 - x - 1$, find x_0 equivalent to $g(x) = x^2 - 1$ find x_1 (guess 10 fixed point iteration is infinite)

$g(x) = \text{sqrt}(x + 1)$ find x (1.32)

Midterm

Friday, October 17th, 2:30pm - 4:20 (110 minutes)

Content(chapters 1-12 + in class)

Variables, expressions, loops, conditions

int, floats, doubles

functions, arrays

Horner, Euclid, Bisection, Fixed point

Structures

e.g Struture for time of day (24-hour time)

```
struct tod{
    int hours;
    int mins;
}

//declaration of struture with tag

int main(void){
    struct tod now = {16,56};
    struct tod later;

    later.hours = 17;
    later.mins = 0;

    printf("%d:%d\n", now.hours, now.mins); //16:50
    printf("%d:%d\n", later.hours, later.mins); //17:0
}
```

```
void todPrint(struct tod when){
    printf("%02d:%02d", when.hours,when.minutes);
}

struct tod todAddTime(struct tod when, int hours, int mins){
    when.mins += mins;
    when.hours += hours + (when.mins) /60;
    when.minss %= 60;
    when.hours %= 24;

    return when;
}

int main(void){
    struct tod later = {18,0};
    todPrint(later); //call by value
    printf("\n");

    return 0;
}
```

Complex Numbers

It is a number with two parts: $x + yi$

A real and a imaginary part.

i is equal to the square root of -1

addition: $(x + yi) + (u + zi) = (x + u) + (y + z)i$
multiplication: $(x + yi)(w + zi) = (xw + wyi + xzi + yzi^2)$
 $(xw - yz) + (wy + xz)i$

As a structure:

```
struct complex{
    double r,i;
}; //store in "complex.h"
```

```
#include <stdio.h>
#include "complex.h"

struct complex complexCreate(double real, double imaginary);
struct complex complexAdd(struct complex x, struct complex y);
struct complex complexMultiply(struct complex x, struct complex y);
void complexPrint(struct complex x);

int main(void){
    struct complex a,b,c;
    a = complexCreate(1.0, 2.0); // (1 + 2i)
    b = complexCreate(3.0, 4.0); // (3 + 4i)
    c = complexAdd(a,b);         //(4.0 + 6.0i)
    complexPrint(c);
    c = complexMultiply(a,b);
    complexPrint(c);
}
```

```
#include <stdio.h>
#include "complex.h"

struct complex complexCreate(double real, double imaginary){
    struct complex c;
    c.r = real;
    c.i = imaginary;

    return c;
}

struct complex complexAdd(struct complex x, struct complex y){
    return complexCreate(x.r + y.r, x.i + y.i)
}

struct complex complexMultiply(struct complex x, struct complex y){

}

void complexPrint(struct complex x){
    printf("%g + %gi", c.r, c.i);
}
```
