**Лабораторная работа №1. Численное интегрирование.**

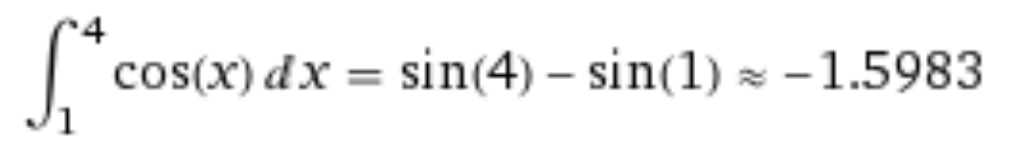
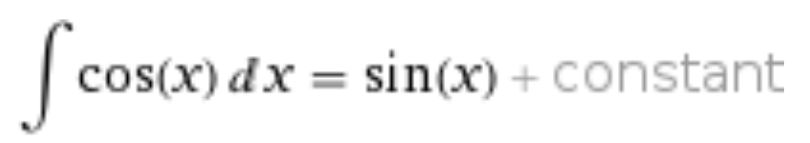
1. Код лабораторной работы написан на языке: С;

Компилятор: онлайн среда Repl.it (clang version 7.0.0)

1. Постановка задачи: Составить программу, которая реализует методы численного интегрирования с постоянным и переменным шагом для интеграла. Программа должна реализовывать следующие меню:



1. Интеграл:



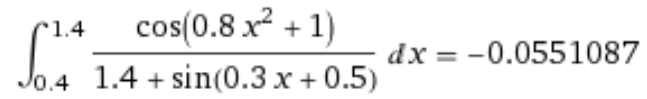
1. Код: <https://repl.it/@sonyadk/labavict>

| Код для табличного | Код для личного |
| --- | --- |
| #include <math.h>  #include <stdlib.h>  #include <stdio.h>  int n;  float a, b, e;  float fun(float x);  float pro(float y);  float pro2(float y);  float pro3(float y);  float fun\_d(float z, float w);  void firstMenu();  void secondMenu();  void thirdMenu();  void lev();  void prav();  void trap();  void par();  void fun\_int\_1();  void fun\_int\_2();  void mainMenu();  void cr\_int();  int main();  float fun(float x){  float y=cos(x);  return y;  }  float fun\_d(float z, float w){  float y = sin(z) + cos(w);  return y;  }  float pro(float y){  float pr = -sin(y);  return pr;  }  float pro2(float y){  float pr = -cos(y);  return pr;  }  float pro3(float y){  float pr = sin(y);  return pr;  }  void cr\_int(){  float c, d, nx, ny, res = 0, iy =0, ix = 0;  printf("\nВведите границы c и d ");  scanf("%f%f", &c, &d);  printf("\nВведите число разбиений nx и ny ");  scanf("%f%f", &nx, &ny);  float nxh = (b - a) / nx;  float nyh = (d - c) / ny;  printf("\nШаг первого: %f\nШаг второго интеграла: %f\n", nxh, nyh);  float sx = 0;  float sy = 0;  for (float i = a; i <= (b - nxh); i+=nxh){  sy = 0;  for (float j = a; j <= (b - nyh); j+=nyh)  {  sy = sy + fabs(fun\_d(i, j));  iy = nyh \* sy;  sx = sx + iy;  ix = nxh \* sx;  }  }  printf("\n > Результат: %.4f\n", ix);  mainMenu();  }  void mainMenu(){  printf("\nВыберите: \n");  printf("1. Вычисление кратного интеграла \n");  printf("2. Вычисление определенного интеграла \n");  printf("3. Выйти из программы \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  cr\_int();  break;  case 2:  firstMenu();  break;  case 3:  printf("\nДо свидания!");  exit(0);  break;  }  }  void firstMenu() {  printf("\nВыберите: \n");  printf("1. Методы с постоянным шагом \n");  printf("2. Методы с переменным шагом \n");  printf("3. Изменить границы \n");  printf("4. Вернуться на уровень выше \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  secondMenu();  break;  case 2:  thirdMenu();  break;  case 3:  mainMenu();  break;  }  }  void secondMenu() {  printf("\nВыберите: \n");  printf("1. Метод левых частей \n");  printf("2. Метод правых частей \n");  printf("3. Метод трапеций \n");  printf("4. Метод парабол \n");  printf("5. Вернуться на уровень выше \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  lev();  break;  case 2:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  prav();  break;  case 3:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  trap();  break;  case 4:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  par();  break;  case 5:  firstMenu();  break;  }  }  void thirdMenu() {  printf("\nВыберите: \n");  printf("1. Метод 1 \n");  printf("2. Метод 2 \n");  printf("3. Вернуться на уровень выше \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  printf("\nВведите точность e ");  scanf("%f", &e);  fun\_int\_1();  break;  case 2:  printf("\nВведите точность e ");  scanf("%f", &e);  fun\_int\_2();  break;  case 3:  firstMenu();  break;  }  }  void prav() {  float h, s, x, res;  h = (b - a) / n;  s = 0;  x = a+h;  while (x <= b)  {  s += fun(x);  x += h;  }  res = h \* s;  printf("\n > Результат методом правых частей: %.4f\n", res);  float r = 0, m = 0;  if (pro(a) > pro(b)) {  m = fabs(pro(a));  }  else{  m = fabs(pro(b));  }  r = (pow(b - a, 2)/(2\*n))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void lev() {  float h, s, x, res;  h = (b - a) / n;  s = 0;  x = a;  while (x <= (b - h))  {  s += fun(x);  x += h;  }  res = h \* s;  printf("\n > Результат методом левых частей: %.4f\n", res);  float r = 0, m = 0;  if (pro(a) > pro(b)) {  m = fabs(pro(a));  }  else{  m = fabs(pro(b));  }  r = (pow(b - a, 2)/(2\*n))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void par() {  float h = 0, s = 0, s1 = 0, s2 = 0, x = 0;  h = (b - a)/n;  s = fun(a) + fun(b);  x = a + h;  while ( x <= (b - h) ){  s1 += fun(x);  x += h \* 2;  }  x = a + h \* 2;  while ( x <= (b - 2 \* h) ) {  s2 += fun(x);  x += 2 \* h;  }  float i = h/3 \* (s + 2 \* s1 + 4 \* s2);  printf("\n > Результат методом парабол: %.4f\n", i);  float r = 0, m = 0;  if (pro3(a) > pro3(b)) {  m = fabs(pro3(a));  }  else{  m = fabs(pro3(b));  }  r = (pow(b - a, 5)/(2880\*pow(n, 4)))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void trap(){  float h = 0, s = 0, x = 0;  h = (b - a)/n;  s = fun(a) + fun(b);  x = a + h;  while ( x <= b){  s += fun(x);  x += h;  }  float i = h \* s;  printf("\n > Результат методом трапеций: %.4f\n", i);  float r = 0, m = 0;  if (pro2(a) > pro2(b)) {  m = fabs(pro2(a));  }  else{  m = fabs(pro2(b));  }  r = (pow(b - a, 3)/(12\*n\*n))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void fun\_int\_1() {  float hv=0, I1=0, I2=0, x = 0, S1 = 0, S2 = 0;  hv = sqrt(e);  float f = (fun(a) + fun(b))/2;  x = a + hv;  while (x <= b){  S1 += fun(x);  x = x + hv;  }  I1 = hv\*(S1+f);  hv = hv/2;  x = a + hv;  while (x <= b){  S2 += fun(x);  x = x + hv;  }  I2 = hv\*(S2+f);  do{  I1 = I2;  hv = hv / 2;  x = a + hv;  S2 = 0;  while (x <= b){  S2 += fun(x);  x = x + hv;  }  I2 = hv\*(S2+f);  } while (fabs(I1-I2) > e);  printf("\nPезультат первым методом: %.4f\n", I2);  thirdMenu();  }  void fun\_int\_2() {  float hs=0, hv=0, I1=0, I2=0, S1=0, S2=0, x = 0;  hv = sqrt(e);  float f = (fun(a) + fun(b))/2;  x = a + hv;  while (x <= b){  S1 += fun(x);  x = x + hv;  }  I1 = hv\*(S1+f);  hs = hv/2;  x = a + hs;  while (x <= b){  S2 += fun(x);  x = x + hv;  }  I2 = hs\*(S2+f);  do{  hv = hs;  I1 = I2;  hs = hv / 2;  x = a + hs;  S1 = 0;  while ( x < b ){  S1 += fun(x);  x = x + hv;  }  I2 = hv \* (f + S1);  }while (fabs(I1-I2)>e);  printf("\nPезультат вторым методом: %.4f\n", I2);  thirdMenu();  }  int main() {  printf("\nВведите границы a и b ");  scanf("%f%f", &a, &b);  mainMenu();  getchar();  return 0;  } | #include <math.h>  #include <stdlib.h>  #include <stdio.h>  int n;  float a, b, e;  float fun(float x);  float pro(float x);  float pro2(float x);  float pro3(float x);  float fun\_d(float z, float w);  void firstMenu();  void secondMenu();  void thirdMenu();  void lev();  void prav();  void trap();  void par();  void fun\_int\_1();  void fun\_int\_2();  void mainMenu();  void cr\_int();  int main();  float fun(float x){  float y=cos(0.8\*x\*x+1)/(1.4 + sin (0.3\*x+0.5)) ;  return y;  }  float fun\_d(float z, float w){  float y = sin(z) + cos(w);  return y;  }  float pro(float x){  float pr = (x\*(-1.6\*sin(0.3\*x + 0.5) - 2.24)\*sin(0.8\*x\*x + 1) - 0.3\*cos(0.3\*x + 0.5)\*cos(0.8\*x\*x + 1))/pow(sin(0.3\*x + 0.5) + 1.4,2);  return pr;  }  float pro2(float x){  float pr = cos(0.8\*x\*2 + 1)\*((0.09\*sin(0.3\*x + 0.5))/pow(sin(0.3\*x + 0.5) + 1.4,2) + (0.18\* pow(cos(0.3\*x + 0.5),2))/pow(sin(0.3\*x + 0.5) + 1.4,3)) + (-1.6\*sin(0.8\*x\*x + 1) - 2.56\*x\*x\* cos(0.8\*x\*x + 1))/(sin(0.3\*x + 0.5) + 1.4) + (0.96\*x\*sin(0.8\*x\*x + 1)\*cos(0.3\*x + 0.5))/pow(sin(0.3\*x + 0.5) + 1.4,2);  return pr;  }  float pro3(float x){  float pr = cos(0.8\*x\*x + 1)\*(-(0.162\*pow(cos(0.3\*x + 0.5), 3))/pow(sin(0.3\*x + 0.5) + 1.4,4) + (0.027\*cos(0.3\*x + 0.5))/pow(sin(0.3\*x + 0.5) + 1.4,2) - (0.162\*sin(0.3\*x + 0.5)\*cos(0.3\*x + 0.5))/pow(sin(0.3\*x + 0.5) + 1.4,3)) - 4.8\*x\*sin(0.8\* x\*x + 1)\*((0.09\*sin(0.3\*x + 0.5))/pow(sin(0.3\*x + 0.5) + 1.4,2) + (0.18\*pow(cos(0.3\*x + 0.5),2))/pow(sin(0.3\*x + 0.5) + 1.4,3) - (0.9\*cos(0.3\*x + 0.5)\*(-1.6\*sin(0.8\*x\*x + 1) - 2.56\*x\*x\*cos(0.8\*x\*x + 1)))/pow(sin(0.3\*x + 0.5) + 1.4,2) + (4.096\*x\*x\*x\*sin(0.8\*x\*x + 1) - 7.68\*x\*cos(0.8\*x\*x + 1))/(sin(0.3\*x + 0.5) + 1.4));  return pr;  }  void cr\_int(){  float c, d, nx, ny, res = 0, iy =0, ix = 0;  printf("\nВведите границы c и d ");  scanf("%f%f", &c, &d);  printf("\nВведите число разбиений nx и ny ");  scanf("%f%f", &nx, &ny);  float nxh = (b - a) / nx;  float nyh = (d - c) / ny;  printf("\nШаг первого: %f\nШаг второго интеграла: %f\n", nxh, nyh);  float sx = 0;  float sy = 0;  for (float i = a; i <= (b - nxh); i+=nxh){  sy = 0;  for (float j = a; j <= (b - nyh); j+=nyh)  {  sy = sy + fabs(fun\_d(i, j));  iy = nyh \* sy;  sx = sx + iy;  ix = nxh \* sx;  }  }  printf("\n > Результат: %.4f\n", ix);  mainMenu();  }  void mainMenu(){  printf("\nВыберите: \n");  printf("1. Вычисление кратного интеграла \n");  printf("2. Вычисление определенного интеграла \n");  printf("3. Выйти из программы \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  cr\_int();  break;  case 2:  firstMenu();  break;  case 3:  printf("\nДо свидания!");  exit(0);  break;  }  }  void firstMenu() {  printf("\nВыберите: \n");  printf("1. Методы с постоянным шагом \n");  printf("2. Методы с переменным шагом \n");  printf("3. Изменить границы \n");  printf("4. Вернуться на уровень выше \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  secondMenu();  break;  case 2:  thirdMenu();  break;  case 3:  mainMenu();  break;  }  }  void secondMenu() {  printf("\nВыберите: \n");  printf("1. Метод левых частей \n");  printf("2. Метод правых частей \n");  printf("3. Метод трапеций \n");  printf("4. Метод парабол \n");  printf("5. Вернуться на уровень выше \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  lev();  break;  case 2:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  prav();  break;  case 3:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  trap();  break;  case 4:  printf("\nВведите количество разбиений ");  scanf("%d", &n);  par();  break;  case 5:  firstMenu();  break;  }  }  void thirdMenu() {  printf("\nВыберите: \n");  printf("1. Метод 1 \n");  printf("2. Метод 2 \n");  printf("3. Вернуться на уровень выше \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  printf("\nВведите точность e ");  scanf("%f", &e);  fun\_int\_1();  break;  case 2:  printf("\nВведите точность e ");  scanf("%f", &e);  fun\_int\_2();  break;  case 3:  firstMenu();  break;  }  }  void prav() {  float h, s, x, res;  h = (b - a) / n;  s = 0;  x = a+h;  while (x <= b)  {  s += fun(x);  x += h;  }  res = h \* s;  printf("\n > Результат методом правых частей: %.4f\n", res);  float r = 0, m = 0;  if (pro(a) > pro(b)) {  m = fabs(pro(a));  }  else{  m = fabs(pro(b));  }  r = (pow(b - a, 2)/(2\*n))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void lev() {  float h, s, x, res;  h = (b - a) / n;  s = 0;  x = a;  while (x <= (b - h))  {  s += fun(x);  x += h;  }  res = h \* s;  printf("\n > Результат методом левых частей: %.4f\n", res);  float r = 0, m = 0;  if (pro(a) > pro(b)) {  m = fabs(pro(a));  }  else{  m = fabs(pro(b));  }  r = (pow(b - a, 2)/(2\*n))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void par() {  float h = 0, s = 0, s1 = 0, s2 = 0, x = 0;  h = (b - a)/n;  s = fun(a) + fun(b);  x = a + h;  while ( x <= (b - h) ){  s1 += fun(x);  x += h \* 2;  }  x = a + h \* 2;  while ( x <= (b - 2 \* h) ) {  s2 += fun(x);  x += 2 \* h;  }  float i = h/3 \* (s + 2 \* s1 + 4 \* s2);  printf("\n > Pезультат методом парабол: %.4f\n", i);  float r = 0, m = 0;  if (pro3(a) > pro3(b)) {  m = fabs(pro3(a));  }  else{  m = fabs(pro3(b));  }  r = (pow(b - a, 5)/(2880\*pow(n, 4)))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void trap(){  float h = 0, s = 0, x = 0;  h = (b - a)/n;  s = fun(a) + fun(b);  x = a + h;  while ( x <= b){  s += fun(x);  x += h;  }  float i = h \* s;  printf("\n > Pезультат методом трапеций: %.4f\n", i);  float r = 0, m = 0;  if (pro2(a) > pro2(b)) {  m = fabs(pro2(a));  }  else{  m = fabs(pro2(b));  }  r = (pow(b - a, 3)/(12\*n\*n))\*m;  printf(" > Остаточный член равен: %f\n", r);  secondMenu();  }  void fun\_int\_1() {  float hv=0, I1=0, I2=0, x = 0, S1 = 0, S2 = 0;  hv = sqrt(e);  float f = (fun(a) + fun(b))/2;  x = a + hv;  while (x <= b){  S1 += fun(x);  x = x + hv;  }  I1 = hv\*(S1+f);  hv = hv/2;  x = a + hv;  while (x <= b){  S2 += fun(x);  x = x + hv;  }  I2 = hv\*(S2+f);  do{  I1 = I2;  hv = hv / 2;  x = a + hv;  S2 = 0;  while (x <= b){  S2 += fun(x);  x = x + hv;  }  I2 = hv\*(S2+f);  } while (fabs(I1-I2) > e);  printf("\nPезультат первым методом: %.4f\n", I2);  thirdMenu();  }  void fun\_int\_2() {  float hs=0, hv=0, I1=0, I2=0, S1=0, S2=0, x = 0;  hv = sqrt(e);  float f = (fun(a) + fun(b))/2;  x = a + hv;  while (x <= b){  S1 += fun(x);  x = x + hv;  }  I1 = hv\*(S1+f);  hs = hv/2;  x = a + hs;  while (x <= b){  S2 += fun(x);  x = x + hv;  }  I2 = hs\*(S2+f);  do{  hv = hs;  I1 = I2;  hs = hv / 2;  x = a + hs;  S1 = 0;  while ( x < b ){  S1 += fun(x);  x = x + hv;  }  I2 = hv \* (f + S1);  }while (fabs(I1-I2)>e);  printf("\nPезультат вторым методом: %.4f\n", I2);  thirdMenu();  }  int main() {  printf("\nВведите границы a и b ");  scanf("%f%f", &a, &b);  mainMenu();  getchar();  return 0;  } |

1. Результаты:

| Метод | Границы | Количество разбиений/точность | Результат |
| --- | --- | --- | --- |
| Левых частей | a = 1  b = 4 | 1000 |  |
| Правых частей |  |
| Парабол |  |
| Трапеций |  |
| Первый алгоритм | 0.01 |  |
| Второй алгоритм |  |

1. Индивидуальный интеграл:



1. Результаты:

| Метод | Границы | Количество разбиений/точность | Результат |
| --- | --- | --- | --- |
| Левых частей | a = 0.4  b = 1.4 | 1000 |  |
| Правых частей |  |
| Парабол |  |
| Трапеций |  |
| Первый алгоритм | 0.001 |  |
| Второй алгоритм |  |

1. Вывод:

В зависимости от того, какую точность желает получить человек, следует использовать различные методы. Так, для получения более точного результата, лучше использовать методы с переменным шагом: они опираются на нужную точность, а не на количество разбиений. Также, на точность влияет остаточный член. Если его не учесть, это приведет к увеличению погрешности и к уменьшению точности.   
Стоит обратить внимание и на особенности языка и компилятора. Различные типы данных в различных языках и компиляторах будут выдавать разные значения, что тоже сказывается на точности вычислений.