

Also, I tested AngelScript with bindings for two cases:

1:

Toggle Spoiler

```
double calculate()
{
    double x, y, sum = 0;
    for(x = 0; x < 1; x += 0.001)
        for(y = 0; y < 1; y += 0.001)
            sum += 1 / (1 - x * y + x - y);
    return sum;
}
```

```
CONSOLE_APP_MAIN
```

```
{
// ...
// Create the AngelScript engine
asIScriptEngine *engine = asCreateScriptEngine(ANGELSCRIPT_VERSION);
if(engine == 0)
{
    Cout() << "Failed to create AngelScript engine.\n";
    SetExitCode(1);
    return;
}
```

```
engine->RegisterGlobalFunction("double calculate()", asFUNCTION(calculate),
asCALL_CDECL);
const String script =
"double compute() {\n"
" return calculate();\n"
"}";
```

```
asIScriptModule *mod = engine->GetModule("script", asGM_ALWAYS_CREATE);
if(mod->AddScriptSection("script", ~script, script.GetCount()) < 0) {
    Cout() << "AddScriptSection() failed\n";
    engine->Release();
    SetExitCode(1);
    return;
}
```

```
if(mod->Build() < 0) {
    Cout() << "Build() failed\n";
    engine->Release();
    SetExitCode(1);
    return;
}
```

```

}

// Create a context that will execute the script.
asIScriptContext *ctx = engine->CreateContext();
if(ctx == 0) {
    Cout() << "Failed to create the context.\n";
    ctx->Release();
    engine->Release();
    SetExitCode(1);
    return;
}

// Find the function id for the function we want to execute.
int funcId = engine->GetModule("script")->GetFunctionIdByDecl("double compute()");
if(funcId < 0) {
    Cout() << "The function \"compute\" wasn't found.\n";
    ctx->Release();
    engine->Release();
    SetExitCode(1);
    return;
}

{
    RTIMING("AngelScript (interpreted with binding)");
    double sum = 0;
    if(ctx->Prepare(funcId) < 0) {
        Cout() << "Failed to prepare the context.\n";
        SetExitCode(1);
        ctx->Release();
        engine->Release();
        return;
    }

    if(ctx->Execute() == asEXECUTION_FINISHED)
        sum = ctx->GetReturnDouble();

    RDUMP(sum);
}

ctx->Release();
engine->Release();
}

```

With following results:

```

1 / (1 - x * y + x - y) = -0.01851851852
fn->Execute() = -0.01851851852
sum = 5190404.858

```

```

sum = 5190404.858
sum = 5190404.858
sum = 5190404.858
TIMING AngelScript (interpreted with binding): 16.00 ms - 16.00 ms (16.00 ms / 1 ), min: 16.00
ms, max: 16.00 ms, nesting: 1 - 1
TIMING Direct      : 13.00 ms - 13.00 ms (13.00 ms / 1 ), min: 13.00 ms, max: 13.00 ms, nesting:
1 - 1
TIMING Compiled    : 53.00 ms - 53.00 ms (53.00 ms / 1 ), min: 53.00 ms, max: 53.00 ms,
nesting: 1 - 1
TIMING Interpreted  : 800.00 ms - 800.00 ms (800.00 ms / 1 ), min: 800.00 ms, max: 800.00 ms,
nesting: 1 - 1

```

```

2:
Toggle Spoiler
double calculate(double x, double y)
{
    return 1 / (1 - x * y + x - y);
}

```

```

CONSOLE_APP_MAIN
{
// ...
// Create the AngelScript engine
asIScriptEngine *engine = asCreateScriptEngine(ANGELSCRIPT_VERSION);
if(engine == 0)
{
    Cout() << "Failed to create AngelScript engine.\n";
    SetExitCode(1);
    return;
}

```

```

engine->RegisterGlobalFunction("double calculate(double, double)", asFUNCTION(calculate),
asCALL_CDECL);

```

```

const String script =
    "double compute() {\n"
    "    double x, y, sum = 0;\n"
    "    for (x = 0; x < 1; x += 0.001)\n"
    "        for (y = 0; y < 1; y += 0.001)\n"
    "            sum += calculate(x, y);\n"
    "    return sum;\n"
    "}";

```

```

asIScriptModule *mod = engine->GetModule("script", asGM_ALWAYS_CREATE);
if(mod->AddScriptSection("script", ~script, script.GetCount()) < 0) {
    Cout() << "AddScriptSection() failed\n";
    engine->Release();
    SetExitCode(1);
}

```

```

    return;
}

if(mod->Build() < 0) {
    Cout() << "Build() failed\n";
    engine->Release();
    SetExitCode(1);
    return;
}

// Create a context that will execute the script.
asIScriptContext *ctx = engine->CreateContext();
if(ctx == 0) {
    Cout() << "Failed to create the context.\n";
    ctx->Release();
    engine->Release();
    SetExitCode(1);
    return;
}

// Find the function id for the function we want to execute.
int funcId = engine->GetModule("script")->GetFunctionIdByDecl("double compute()");
if(funcId < 0) {
    Cout() << "The function \"compute\" wasn't found.\n";
    ctx->Release();
    engine->Release();
    SetExitCode(1);
    return;
}

{
    RTIMING("AngelScript (fully interpreted with binding)");
    double sum = 0;
    if(ctx->Prepare(funcId) < 0) {
        Cout() << "Failed to prepare the context.\n";
        SetExitCode(1);
        ctx->Release();
        engine->Release();
        return;
    }

    if(ctx->Execute() == asEXECUTION_FINISHED)
        sum = ctx->GetReturnDouble();

    RDUMP(sum);
}

ctx->Release();

```

```
engine->Release();  
}
```

With following results:

$1 / (1 - x * y + x - y) = -0.01851851852$

fn->Execute() = -0.01851851852

sum = 5190404.858

sum = 5190404.858

sum = 5190404.858

sum = 5190404.858

TIMING AngelScript (fully interpreted with binding): 170.00 ms - 170.00 ms (170.00 ms / 1), min: 170.00 ms, max: 170.00 ms, nesting: 1 - 1

TIMING Direct : 14.00 ms - 14.00 ms (14.00 ms / 1), min: 14.00 ms, max: 14.00 ms, nesting: 1 - 1

TIMING Compiled : 54.00 ms - 54.00 ms (54.00 ms / 1), min: 54.00 ms, max: 54.00 ms, nesting: 1 - 1

TIMING Interpreted : 793.00 ms - 793.00 ms (793.00 ms / 1), min: 793.00 ms, max: 793.00 ms, nesting: 1 - 1
