```
polly.par.setup:
 %polly.par.LBPtr = alloca i64
 %polly.par.UBPtr = alloca i64
 %polly.par.userContext1 = bitcast i8* %polly.par.userContext to { i64, i32,
... i64, double*, double*, double* }*
 %0 = getelementptr inbounds { i64, i32, i64, double*, double*, double* }, {
... i64, i32, i64, double*, double*, double* }* %polly.par.userContext1, i32 0,
... i32 1
 %polly.subfunc.arg.n = load i32, i32* %0
 %1 = getelementptr inbounds { i64, i32, i64, double*, double*, double* }, {
... i64, i32, i64, double*, double*, double* }* %polly.par.userContext1, i32 0,
... i32 2
 %polly.subfunc.arg. = load i64, i64* %1
 %2 = getelementptr inbounds { i64, i32, i64, double*, double*, double* }, {
... i64, i32, i64, double*, double*, double* }* %polly.par.userContext1, i32 0,
... i32 3
 %polly.subfunc.arg.x1 = load double*, double** %2
 %3 = getelementptr inbounds { i64, i32, i64, double*, double*, double* }, {
... i64, i32, i64, double*, double*, double* }* %polly.par.userContext1, i32 0,
... i32 4
 %polly.subfunc.arg.A = load double*, double** %3
 %4 = getelementptr inbounds { i64, i32, i64, double*, double*, double* }, {
... i64, i32, i64, double*, double*, double* }* %polly.par.userContext1, i32 0,
... i32 5
 %polly.subfunc.arg.y_1 = load double*, double** %4
 br label %polly.par.checkNext
```

```
polly.par.checkNext:
 %5 = call i8 @GOMP_loop_runtime_next(i64* %polly.par.LBPtr, i64*
... %polly.par.UBPtr)
 %6 = icmp ne i8 %5, 0
 br i1 %6, label %polly.par.loadIVBounds, label %polly.par.exit
```

```
polly.par.loadIVBounds:
 %polly.par.LB = load i64, i64* %polly.par.LBPtr
 %polly.par.UB = load i64, i64* %polly.par.UBPtr
 %polly.par.UBAdjusted = sub i64 %polly.par.UB, 1
 br label %polly.loop_preheader
```

```
polly.par.exit:
 call void @GOMP_loop_end_nowait()
 ret void
```

```
polly.loop_preheader:
 %7 = zext i32 %polly.subfunc.arg.n to i64
 br label %polly.loop_header
```

```
polly.loop_header:
 %polly.indvar = phi i64 [ %polly.par.LB, %polly.loop_preheader ], [
... %polly.indvar_next, %polly.loop_exit4 ]
 br label %polly.loop_preheader3
```

```
polly.loop_preheader3:
 br label %polly.loop_header2
```

```
polly.loop_header2:
 %polly.indvar5 = phi i64 [ 0, %polly.loop_preheader3 ], [
... %polly.indvar_next6, %polly.loop_exit10 ]
 %8 = mul nsw i64 32, %polly.indvar
 %9 = sub nsw i64 undef, %8
 %10 = sub nsw i64 %9, 1
 %11 = icmp slt i64 31, %10
 %12 = select i1 %11, i64 31, i64 %10
 br label %polly.loop_if
```

```
polly.loop_if:
 %polly.loop_guard = icmp sle i64 0, %12
 br i1 %polly.loop_guard, label %polly.loop_preheader9, label
... %polly.loop_exit10
```

```
polly.loop_preheader9:
 br label %polly.loop_header8
```

```
polly.loop_header8:
 %polly.indvar11 = phi i64 [ 0, %polly.loop_preheader9 ], [
... %polly.indvar_next12, %polly.loop_exit18 ]
 %13 = mul nsw i64 32, %polly.indvar5
 %14 = sub nsw i64 undef, %13
 %15 = sub nsw i64 %14, 1
 %16 = icmp slt i64 31, %15
 %17 = select i1 %16, i64 31, i64 %15
 br label %polly.loop_if15
```

```
polly.loop_if15:
 %polly.loop_guard19 = icmp sle i64 0, %17
 br i1 %polly.loop_guard19, label %polly.loop_preheader17, label
... %polly.loop_exit18
```

```
polly.loop_preheader17:
 br label %polly.loop_header16
```

```
polly.loop_header16:
 %polly.indvar20 = phi i64 [ 0, %polly.loop_preheader17 ], [
... %polly.indvar_next21, %polly.stmt.for.body8 ]
 %18 = mul nsw i64 32, %polly.indvar
 %19 = add nsw i64 %18, %polly.indvar11
 %20 = mul nsw i64 32, %polly.indvar5
 %21 = add nsw i64 %20, %polly.indvar20
 br label %polly.stmt.for.body8
```

```
polly.stmt.for.body8:
 %scevgep = getelementptr double, double* %polly.subfunc.arg.x1, i64 %19
 %_p_scalar_ = load double, double* %scevgep, align 8, !alias.scope !1,
... !noalias !3
 %22 = mul i64 %7, %19
 %23 = add i64 %21, %22
 %scevgep24 = getelementptr double, double* %polly.subfunc.arg.A, i64 %23
 %_p_scalar_25 = load double, double* %scevgep24, align 8, !alias.scope !7,
... !noalias !8
 %scevgep26 = getelementptr double, double* %polly.subfunc.arg.y_1, i64 %21
 %_p_scalar_27 = load double, double* %scevgep26, align 8, !alias.scope !4,
... !noalias !9
 %p_mul = fmul double %_p_scalar_25, %_p_scalar_27
 %p_add15 = fadd double %_p_scalar_, %p_mul
 %scevgep28 = getelementptr double, double* %polly.subfunc.arg.x1, i64 %19
 store double %p_add15, double* %scevgep28, align 8, !alias.scope !1,
... !noalias !3
 %24 = trunc i64 %21 to i32
 %25 = add i32 %24, 1
 %p_exitcond17 = icmp ne i32 %25, %polly.subfunc.arg.n
 %polly.indvar_next21 = add nsw i64 %polly.indvar20, 1
 %polly.adjust_ub22 = sub i64 %17, 1
 %polly.loop_cond23 = icmp sle i64 %polly.indvar20, %polly.adjust_ub22
 br i1 %polly.loop_cond23, label %polly.loop_header16, label
... %polly.loop_exit18
```

```
polly.loop_exit18:
 %polly.indvar_next12 = add nsw i64 %polly.indvar11, 1
 %polly.adjust_ub13 = sub i64 %12, 1
 %polly.loop_cond14 = icmp sle i64 %polly.indvar11, %polly.adjust_ub13
 br i1 %polly.loop_cond14, label %polly.loop_header8, label %polly.loop_exit10
```

```
polly.loop_exit10:
 %polly.indvar_next6 = add nsw i64 %polly.indvar5, 1
 %polly.loop_cond7 = icmp sle i64 %polly.indvar5, -1
 br i1 %polly.loop_cond7, label %polly.loop_header2, label %polly.loop_exit4
```

```
polly.loop_exit4:
 %polly.indvar_next = add nsw i64 %polly.indvar, 1
 %polly.adjust_ub = sub i64 %polly.par.UBAdjusted, 1
 %polly.loop_cond = icmp sle i64 %polly.indvar, %polly.adjust_ub
 br i1 %polly.loop_cond, label %polly.loop_header, label %polly.loop_exit
```

```
polly.loop_exit:
 br label %polly.par.checkNext
```

CFG for 'polly_task_polly_subfn' function