

This document outlines a step-by-step plan for adding single-precision C implementations to ``stdlib/math/base/special``. The implementation is structured in phases, ensuring that high-priority functions are developed first. Each phase (except phase 0 and phase 1) builds upon the previous one, prioritizing functions with fewer dependencies to enable faster package expansion.

For ease, all implementations are arranged to follow a logical top-to-bottom progression.

Phase 0: Standalone Packages

This stage includes independent packages that have no prerequisites and are not prerequisites for other functions. Since they do not depend on or impact other implementations, they can be developed at any time without affecting the progression of later stages.

- `bernoullif`
- `binomcoeff`
- `cinvf`
- `csignumf`
- `erfcinvf`
- `erfinvf`
- `factorial2f`
- `fibonaccif`
- `fibonacci-indexf`
- `frexpf`
- `gamma-lanczos-sum-expg-scaledf`
- `gammaasgnf`
- `logitf`
- `lucasf`
- `minmaxf`
- `minmaxabsf`
- `modff`
- `negafibonaccif`
- `spencef`
- `tribonaccif`
- `wrapf`

Phase 1: Mostly Independent Packages

This stage consists of functions that are largely independent, like Stage 0, but with minimal interdependencies. These functions can still be implemented at any time without significantly impacting later phases.

Package Name	Missing Prerequisite(s)
atan2f	
cphasef	atan2f
cpolarf	cphasef
ellipkf	
ellipef	ellipkf

Phase 2: Trigonometric

This is where the real fun begins! This phase is mostly made up of trigonometric functions or ones that depend on them. A lot of these rely on `rempio2f` and the sine, cosine, and tangent kernels. But once those are implemented, they will unlock a huge chunk of single-precision packages, pushing the project forward in a big way.

Package Name	Missing Prerequisite(s)
<code>rempio2f</code>	
<code>kernel-cosdf</code>	
<code>kernel-sindf</code>	
<code>sinf</code>	<code>kernel-sindf</code> , <code>kernel-cosdf</code> , <code>rempio2f</code>
<code>covercosf</code>	<code>sinf</code>
<code>coversinf</code>	<code>sinf</code>
<code>cscf</code>	<code>sinf</code>
<code>gammaf</code>	<code>sinf</code>
<code>factorialf</code>	<code>gammaf</code>
<code>hacovercosf</code>	<code>sinf</code>
<code>hacoversinf</code>	<code>sinf</code>
<code>cosf</code>	<code>kernel-cosdf</code> , <code>kernel-sindf</code> , <code>rempio2f</code>
<code>cosm1f</code>	<code>cosf</code>
<code>havercosf</code>	<code>cosf</code>
<code>haversinf</code>	<code>cosf</code>
<code>secf</code>	<code>cosf</code>
<code>vercosf</code>	<code>cosf</code>

versinf	cosf
cospif	sinf, cosf
sicif	sinf, cosf
sinpif	sinf, cosf
sincf	sinpif
trigammaf	sinpif
gamma1nf	sinpif
factorial1nf	gamma1nf
kernel-sincosf	
sincosf	kernel-sincosf, rempio2f
fresnelf	sincosf
fresnelcf	sincosf
fresnelsf	sincosf
sincospif	sincosf
besselj0f	sincosf
besselj1f	sincosf
bessely0f	sincosf besselj0f
bessely1f	sincosf besselj1f
sindf	kernel-cosdf, kernel-sindf
cscdf	sindf

cosdf	kernel-cosdf, kernel-sindf
secdf	cosdf
tandf	sindf, cosdf
cotdf	sindf, cosdf
sincosdf	kernel-sincosf, rempio2f
kernel-tandf	
tanf	kernel-tanf, rempio2f
cotf	tanf
digammaf	tanf

Phase 3: Logarithm

This phase contains fewer packages but plays a “nice” role in setting up the implementations to follow.

Package Name	Missing Prerequisite(s)
kernel-logf	
log10f	kernel-logf
log2f	kernel-logf
log1pf	
xlog1pyf	log1pf
acoshf	log1pf
atanhf	log1pf
asinhf	log1pf
acothf	atanhf
acschf	asinhf
asechf	acoshf
betaln	gammaln, log1pf, gammaf
binomcoefln	betaln

Phase 4: Power

This phase introduces `powf`, a key function that serves as a prerequisite for many other functions. It is placed here after the logarithm phase to avoid repetitive back-and-forth between implementations.

Package Name	Missing Prerequisite(s)
<code>powf</code>	
<code>binetf</code>	<code>cospif</code> , <code>powf</code>
<code>ceilnf</code>	<code>powf</code>
<code>ceilbf</code>	<code>powf</code> , <code>ceilnf</code>
<code>cceilnf</code>	<code>ceilnf</code>
<code>ceil10f</code>	<code>powf</code> , <code>log10f</code>
<code>ceilsdf</code>	<code>powf</code> , <code>log10f</code>
<code>ceil2f</code>	<code>powf</code> , <code>log2f</code>
<code>floornf</code>	<code>powf</code>
<code>cfloornf</code>	<code>floornf</code>
<code>floorbf</code>	<code>powf</code> , <code>floornf</code>
<code>floor10f</code>	<code>powf</code> , <code>log10f</code>
<code>floorsdf</code>	<code>powf</code> , <code>log10f</code>
<code>floor2f</code>	<code>powf</code> , <code>log2f</code>
<code>roundnf</code>	<code>powf</code>
<code>croundnf</code>	<code>roundnf</code>
<code>roundbf</code>	<code>powf</code> , <code>roundnf</code>
<code>round10f</code>	<code>powf</code> , <code>log10f</code>

roundsdf	powf, log10f
round2f	powf, log2f
truncnf	powf
truncbf	powf, truncnf
trunc10f	powf, log10f
truncsdf	powf, log10f
trunc2f	powf, log2f

Phase 5: Exponential

Package Name	Missing Prerequisite(s)
expf	
expitf	expf
erff	expf
erfcf	expf
erfcxf	expf
sinhf	expf
cschf	sinhf
coshf	expf
tanhf	expf
cothf	tanhf
ellipjf	sinhf, coshf, tanhf, sinf, cosf
betaf	log1pf, expf, powf
ccisf	expf, sincosf
cexpf	sincosf, expf
gammaincf	gamma1nf, expf, powf
log1pexpf	log1pf, expf
logaddexpf	log1pf, expf
riemann-zetaf	expf, gammaf, gamma1nf, sinpif, powf
polygammaf	trigammaf, digammaf, riemann-zetaf

gamma-lanczos-sumf	
gamma-delta-ratiof	gamma-lanczos-sumf, gammaf, log1pf, expf, powf
falling-factorialf	gamma-delta-ratiof
rising-factorialf	gamma-delta-ratiof, falling-factorialf

Phase 6: Extended Exponential

Package Name	Missing Prerequisite(s)
expm1f	
expm1relf	expm1f
boxcoxf	expm1f
boxcox1pf	expm1f, log1pf
boxcox1pinvf	expm1f, log1pf
boxcoxinvf	expm1f, log1pf
gamma1pm1f	expm1f, log1pf
sqrt1pm1f	expm1f, log1pf
log1mexpf	expm1f, expf, log1pf
powm1f	expm1f, powf
dirichlet-etaf	powm1f, riemann-zetaf
kernel-betaincf	expm1f, log1pf, expf, powf
betaincf	kernel-betaincf
kernel-betaincinvf	betaincf, expm1f, log1pf, expf, powf
betaincinvf	kernel-betaincinvf
gammamaincinvf	gamma1lnf, erfcinvf, gammaf, powf

Phase 7: The Side Quest

These implementations aren't necessarily harder than the previous ones, but they offer less immediate return on investment (I think!). Since their timing doesn't matter much, they can be worked on whenever.

Package Name	Missing Prerequisite(s)
exp10f	
exp2f	
log1pmxf	